

Kompilátory

Cvičenie 3: Syntaktická analýza zhora nadol

Peter Kostolányi

24. októbra 2017

Oznamy

Do 6. novembra 2017 treba odovzdať neformálnu špecifikáciu projektu:

Oznamy

Do 6. novembra 2017 treba odovzdať neformálnu špecifikáciu projektu:

- ▶ Odovzdáva sa mailom

Oznamy

Do 6. novembra 2017 treba odovzdať neformálnu špecifikáciu projektu:

- ▶ Odovzdáva sa mailom
- ▶ Podrobné zadanie je na stránke predmetu

Oznamy

Do 6. novembra 2017 treba odovzdať neformálnu špecifikáciu projektu:

- ▶ Odovzdáva sa mailom
- ▶ Podrobné zadanie je na stránke predmetu
- ▶ Za neskoré odovzdanie je penalizácia –10% z celkového bodového zisku z cvičení

Oznamy

Do **6. novembra 2017** treba odovzdať neformálnu špecifikáciu projektu:

- ▶ Odovzdáva sa mailom
- ▶ Podrobné zadanie je na stránke predmetu
- ▶ Za neskoré odovzdanie je penalizácia –10% z celkového bodového zisku z cvičení

Do **21. novembra 2017** treba odovzdať riešenia prvej sady domácich úloh:

Oznamy

Do **6. novembra 2017** treba odovzdať neformálnu špecifikáciu projektu:

- ▶ Odovzdáva sa mailom
- ▶ Podrobné zadanie je na stránke predmetu
- ▶ Za neskoré odovzdanie je penalizácia —10% z celkového bodového zisku z cvičení

Do **21. novembra 2017** treba odovzdať riešenia prvej sady domácich úloh:

- ▶ Odovzdáva sa mailom, alebo osobne na cvičení resp. na M-258

Oznamy

Do **6. novembra 2017** treba odovzdať neformálnu špecifikáciu projektu:

- ▶ Odovzdáva sa mailom
- ▶ Podrobné zadanie je na stránke predmetu
- ▶ Za neskoré odovzdanie je penalizácia —10% z celkového bodového zisku z cvičení

Do **21. novembra 2017** treba odovzdať riešenia prvej sady domácich úloh:

- ▶ Odovzdáva sa mailom, alebo osobne na cvičení resp. na M-258
- ▶ Rukou písané riešenia, prosím, odovzdajte osobne

Oznamy

Do **6. novembra 2017** treba odovzdať neformálnu špecifikáciu projektu:

- ▶ Odovzdáva sa mailom
- ▶ Podrobné zadanie je na stránke predmetu
- ▶ Za neskoré odovzdanie je penalizácia —10% z celkového bodového zisku z cvičení

Do **21. novembra 2017** treba odovzdať riešenia prvej sady domácich úloh:

- ▶ Odovzdáva sa mailom, alebo osobne na cvičení resp. na M-258
- ▶ Rukou písané riešenia, prosím, odovzdajte osobne
- ▶ Zadania sa objavia na stránke po tomto cvičení

Oznamy

Do **6. novembra 2017** treba odovzdať neformálnu špecifikáciu projektu:

- ▶ Odovzdáva sa mailom
- ▶ Podrobné zadanie je na stránke predmetu
- ▶ Za neskoré odovzdanie je penalizácia —10% z celkového bodového zisku z cvičení

Do **21. novembra 2017** treba odovzdať riešenia prvej sady domácich úloh:

- ▶ Odovzdáva sa mailom, alebo osobne na cvičení resp. na M-258
- ▶ Rukou písané riešenia, prosím, odovzdajte osobne
- ▶ Zadania sa objavia na stránke po tomto cvičení
- ▶ Dve teoretické úlohy po 3 body (k témam z dnešného cvičenia)

Oznamy

Do **6. novembra 2017** treba odovzdať neformálnu špecifikáciu projektu:

- ▶ Odovzdáva sa mailom
- ▶ Podrobné zadanie je na stránke predmetu
- ▶ Za neskoré odovzdanie je penalizácia –10% z celkového bodového zisku z cvičení

Do **21. novembra 2017** treba odovzdať riešenia prvej sady domácich úloh:

- ▶ Odovzdáva sa mailom, alebo osobne na cvičení resp. na M-258
- ▶ Rukou písané riešenia, prosím, odovzdajte osobne
- ▶ Zadania sa objavajú na stránke po tomto cvičení
- ▶ Dve teoretické úlohy po 3 body (k témam z dnešného cvičenia)
- ▶ 21. novembra bude zverejnená druhá (prakticky zameraná) sada úloh

Časť 1:

Základné princípy syntaktickej analýzy zhora nadol

Úloha syntaktickej analýzy pri tvorbe kompilátorov

Prúd vstupných
symbolov

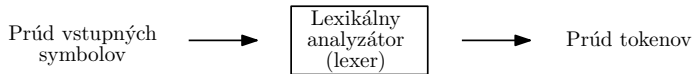
Úloha syntaktickej analýzy pri tvorbe kompilátorov

Prúd vstupných
symbolov

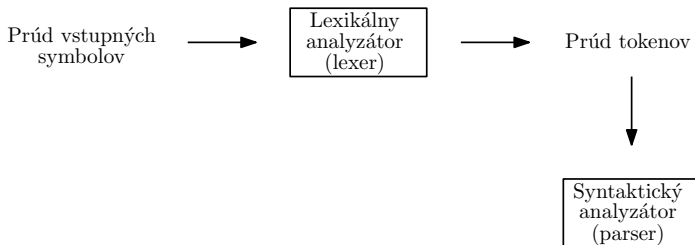


Lexikálny
analyzátor
(lexer)

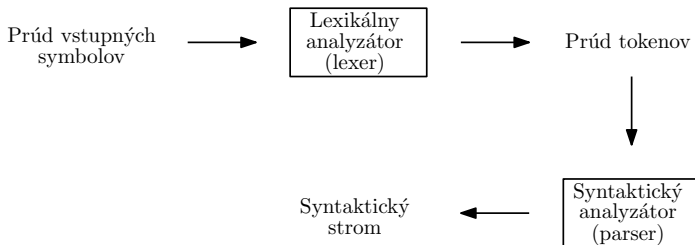
Úloha syntaktickej analýzy pri tvorbe kompilátorov



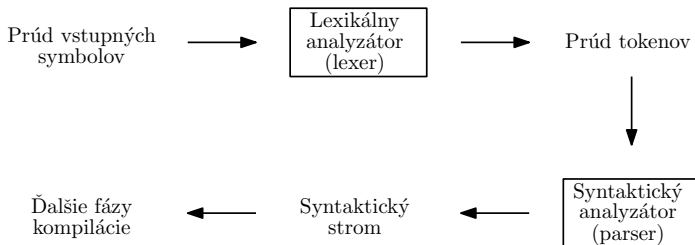
Úloha syntaktickej analýzy pri tvorbe kompilátorov



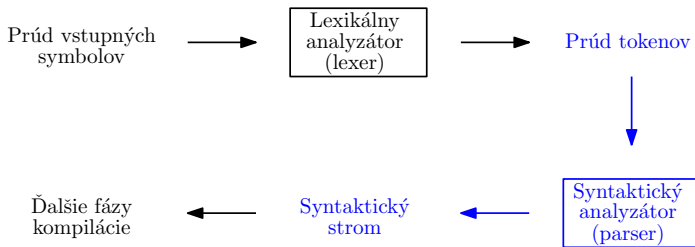
Úloha syntaktickej analýzy pri tvorbe kompilátorov



Úloha syntaktickej analýzy pri tvorbe kompilátorov



Úloha syntaktickej analýzy pri tvorbe kompilátorov



Úloha syntaktickej analýzy pri tvorbe kompilátorov

Syntaktický analyzátor (parser) pre jazyk generovaný gramatikou G :

Úloha syntaktickej analýzy pri tvorbe kompilátorov

Syntaktický analyzátor (parser) pre jazyk generovaný gramatikou G :

- ▶ Číta prúd tokenov generovaný lexikálnym analyzátorom

Úloha syntaktickej analýzy pri tvorbe kompilátorov

Syntaktický analyzátor (parser) pre jazyk generovaný gramatikou G :

- ▶ Číta prúd tokenov generovaný lexikálnym analyzátorom
- ▶ Overuje, či vstupná postupnosť tokenov patrí do $L(G)$

Úloha syntaktickej analýzy pri tvorbe kompilátorov

Syntaktický analyzátor (parser) pre jazyk generovaný gramatikou G :

- ▶ Číta prúd tokenov generovaný lexikálnym analyzátorom
- ▶ Overuje, či vstupná postupnosť tokenov patrí do $L(G)$
- ▶ **Ak áno:** explicitne alebo implicitne skonštruuje strom odvodu vstupnej postupnosti v G (niekedy stačí vykonať vhodné akcie)

Úloha syntaktickej analýzy pri tvorbe kompilátorov

Syntaktický analyzátor (parser) pre jazyk generovaný gramatikou G :

- ▶ Číta prúd tokenov generovaný lexikálnym analyzátorom
- ▶ Overuje, či vstupná postupnosť tokenov patrí do $L(G)$
- ▶ **Ak áno:** explicitne alebo implicitne skonštruuje strom odvodu vstupnej postupnosti v G (niekedy stačí vykonať vhodné akcie)
- ▶ **Ak nie:** vyhlási vhodnú syntaktickú chybu a snaží sa z nej zotaviť

Úloha syntaktickej analýzy pri tvorbe kompilátorov

Syntaktický analyzátor (parser) pre jazyk generovaný gramatikou G :

- ▶ Číta prúd tokenov generovaný lexikálnym analyzátorom
- ▶ Overuje, či vstupná postupnosť tokenov patrí do $L(G)$
- ▶ **Ak áno:** explicitne alebo implicitne skonštruuje strom odvodu vstupnej postupnosti v G (niekedy stačí vykonať vhodné akcie)
- ▶ **Ak nie:** vyhlási vhodnú syntaktickú chybu a snaží sa z nej zotaviť

Gramatika G popisujúca vstupný jazyk je zvyčajne **bezkontextová**:

Úloha syntaktickej analýzy pri tvorbe kompilátorov

Syntaktický analyzátor (parser) pre jazyk generovaný gramatikou G :

- ▶ Číta prúd tokenov generovaný lexikálnym analyzátorom
- ▶ Overuje, či vstupná postupnosť tokenov patrí do $L(G)$
- ▶ **Ak áno:** explicitne alebo implicitne skonštruuje strom odvodu vstupnej postupnosti v G (niekedy stačí vykonať vhodné akcie)
- ▶ **Ak nie:** vyhlási vhodnú syntaktickú chybu a snaží sa z nej zotaviť

Gramatika G popisujúca vstupný jazyk je zvyčajne **bezkontextová**:

- ▶ Vo viacznačných gramatikách môže mať jedno slovo viacero stromov odvodu

Úloha syntaktickej analýzy pri tvorbe kompilátorov

Syntaktický analyzátor (parser) pre jazyk generovaný gramatikou G :

- ▶ Číta prúd tokenov generovaný lexikálnym analyzátorom
- ▶ Overuje, či vstupná postupnosť tokenov patrí do $L(G)$
- ▶ **Ak áno**: explicitne alebo implicitne skonštruuje strom odvodu vstupnej postupnosti v G (niekedy stačí vykonať vhodné akcie)
- ▶ **Ak nie**: vyhlási vhodnú syntaktickú chybu a snaží sa z nej zotaviť

Gramatika G popisujúca vstupný jazyk je zvyčajne **bezkontextová**:

- ▶ Vo viacznačných gramatikách môže mať jedno slovo viacero stromov odvodu
- ▶ Teoreticky významné triedy efektívne parsovatelných gramatík ($LL(k)$, $LR(k)$, ...): len jednoznačné gramatiky

Úloha syntaktickej analýzy pri tvorbe kompilátorov

Syntaktický analyzátor (parser) pre jazyk generovaný gramatikou G :

- ▶ Číta prúd tokenov generovaný lexikálnym analyzátorom
- ▶ Overuje, či vstupná postupnosť tokenov patrí do $L(G)$
- ▶ **Ak áno**: explicitne alebo implicitne skonštruuje strom odvodu vstupnej postupnosti v G (niekedy stačí vykonať vhodné akcie)
- ▶ **Ak nie**: vyhlási vhodnú syntaktickú chybu a snaží sa z nej zotaviť

Gramatika G popisujúca vstupný jazyk je zvyčajne **bezkontextová**:

- ▶ Vo viacznačných gramatikách môže mať jedno slovo viacero stromov odvodu
- ▶ Teoreticky významné triedy efektívne parsovatelných gramatík ($LL(k)$, $LR(k)$, ...): len jednoznačné gramatiky
- ▶ V praxi sa zídu aj viacznačné gramatiky s prioritami pravidiel (väčšina parsovacích metód sa dá rozšíriť na takéto gramatiky)

Syntaktická analýza zhora nadol

Explicitná alebo implicitná konštrukcia stromu odvodenia, ktorá:

Syntaktická analýza zhora nadol

Explicitná alebo implicitná konštrukcia stromu odvodenia, ktorá:

- ▶ Začína v koreni (počiatočný neterminál)

Syntaktická analýza zhora nadol

Explicitná alebo implicitná konštrukcia stromu odvodenia, ktorá:

- ▶ Začína v koreni (počiatočný neterminál)
- ▶ Postupne vytvára ďalšie uzly stromu smerom nadol až k jeho listom („prehľadávaním do hĺbky“)

Syntaktická analýza zhora nadol (příklad)

Uvažujme bezkontextovou gramatiku s následujícími pravidly:

Syntaktická analýza zhora nadol (příklad)

Uvažujme bezkontextovou gramatiku s následujícími pravidly:

$$\sigma \rightarrow \alpha c \beta \mid \beta d \alpha$$

$$\alpha \rightarrow a \alpha \mid b b \alpha \mid \varepsilon$$

$$\beta \rightarrow a a \beta \mid \beta b \mid \varepsilon$$

Syntaktická analýza zhora nadol (príklad)

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow \alpha c \beta \mid \beta d \alpha$$

$$\alpha \rightarrow a \alpha \mid b b \alpha \mid \varepsilon$$

$$\beta \rightarrow a a \beta \mid \beta b \mid \varepsilon$$

Konštrukcia syntaktického stromu k slovu *aabbacaab*:

Syntaktická analýza zhora nadol (príklad)

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow \alpha c \beta \mid \beta d \alpha$$

$$\alpha \rightarrow a \alpha \mid b b \alpha \mid \varepsilon$$

$$\beta \rightarrow a a \beta \mid \beta b \mid \varepsilon$$

Konštrukcia syntaktického stromu k slovu *aabbacaab*:

σ

Syntaktická analýza zhora nadol (príklad)

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow \alpha c \beta \mid \beta d \alpha$$

$$\alpha \rightarrow a \alpha \mid b b \alpha \mid \varepsilon$$

$$\beta \rightarrow a a \beta \mid \beta b \mid \varepsilon$$

Konštrukcia syntaktického stromu k slovu *aabbacaab*:

σ

Syntaktická analýza zhora nadol (príklad)

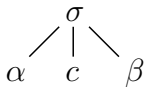
Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow \alpha c \beta \mid \beta d \alpha$$

$$\alpha \rightarrow a \alpha \mid b b \alpha \mid \varepsilon$$

$$\beta \rightarrow a a \beta \mid \beta b \mid \varepsilon$$

Konštrukcia syntaktického stromu k slovu *aabbacaab*:



Syntaktická analýza zhora nadol (príklad)

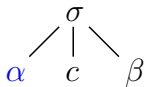
Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow \alpha c \beta \mid \beta d \alpha$$

$$\alpha \rightarrow a \alpha \mid b b \alpha \mid \varepsilon$$

$$\beta \rightarrow a a \beta \mid \beta b \mid \varepsilon$$

Konštrukcia syntaktického stromu k slovu *aabbacaab*:



Syntaktická analýza zhora nadol (príklad)

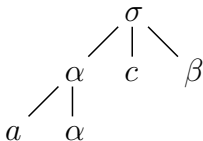
Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow \alpha c \beta \mid \beta d \alpha$$

$$\alpha \rightarrow a \alpha \mid b b \alpha \mid \varepsilon$$

$$\beta \rightarrow a a \beta \mid \beta b \mid \varepsilon$$

Konštrukcia syntaktického stromu k slovu *aabbacaab*:



Syntaktická analýza zhora nadol (príklad)

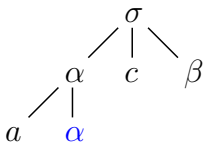
Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow \alpha c \beta \mid \beta d \alpha$$

$$\alpha \rightarrow a \alpha \mid b b \alpha \mid \varepsilon$$

$$\beta \rightarrow a a \beta \mid \beta b \mid \varepsilon$$

Konštrukcia syntaktického stromu k slovu *aabbacaab*:



Syntaktická analýza zhora nadol (příklad)

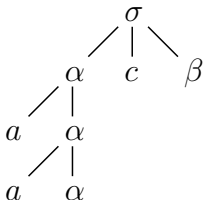
Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow \alpha c \beta \mid \beta d \alpha$$

$$\alpha \rightarrow a\alpha \mid b\alpha \mid \varepsilon$$

$$\beta \rightarrow aa\beta \mid \beta b \mid \varepsilon$$

Konstrukcia syntaktického stromu k slovu *aabbacaab*:



Syntaktická analýza zhora nadol (příklad)

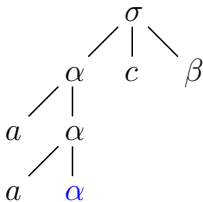
Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow \alpha c \beta \mid \beta d \alpha$$

$$\alpha \rightarrow a\alpha \mid b\alpha \mid \varepsilon$$

$$\beta \rightarrow aa\beta \mid \beta b \mid \varepsilon$$

Konstrukcia syntaktického stromu k slovu *aabbacaab*:



Syntaktická analýza zhora nadol (příklad)

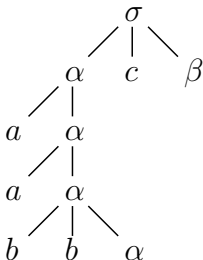
Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow \alpha c \beta \mid \beta d \alpha$$

$$\alpha \rightarrow a\alpha \mid b\alpha \mid \varepsilon$$

$$\beta \rightarrow aa\beta \mid \beta b \mid \varepsilon$$

Konstrukcia syntaktického stromu k slovu *aabbacaab*:



Syntaktická analýza zhora nadol (príklad)

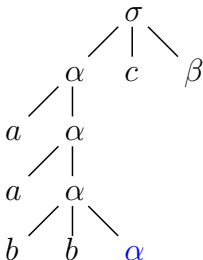
Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow \alpha c \beta \mid \beta d \alpha$$

$$\alpha \rightarrow a \alpha \mid b b \alpha \mid \varepsilon$$

$$\beta \rightarrow a a \beta \mid \beta b \mid \varepsilon$$

Konštrukcia syntaktického stromu k slovu *aabbacaab*:



Syntaktická analýza zhora nadol (príklad)

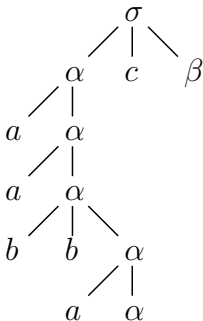
Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow \alpha c \beta \mid \beta d \alpha$$

$$\alpha \rightarrow a \alpha \mid b b \alpha \mid \varepsilon$$

$$\beta \rightarrow a a \beta \mid \beta b \mid \varepsilon$$

Konštrukcia syntaktického stromu k slovu *aabbacaab*:



Syntaktická analýza zhora nadol (příklad)

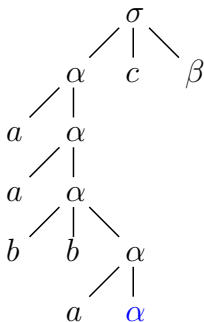
Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow \alpha c \beta \mid \beta d \alpha$$

$$\alpha \rightarrow a\alpha \mid b\alpha \mid \varepsilon$$

$$\beta \rightarrow \mathbf{aa}\beta \mid \beta \mathbf{b} \mid \varepsilon$$

Konstrukcia syntaktického stromu k slovu *aabbacaab*:



Syntaktická analýza zhora nadol (príklad)

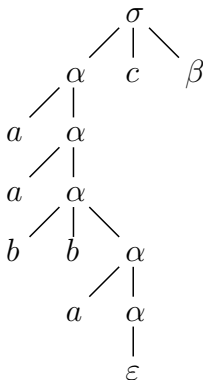
Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow \alpha c \beta \mid \beta d \alpha$$

$$\alpha \rightarrow a \alpha \mid b b \alpha \mid \varepsilon$$

$$\beta \rightarrow a a \beta \mid \beta b \mid \varepsilon$$

Konštrukcia syntaktického stromu k slovu *aabbacaab*:



Syntaktická analýza zhora nadol (príklad)

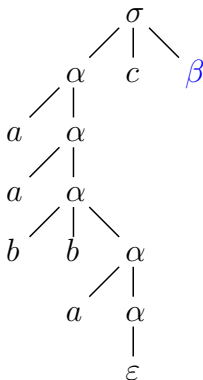
Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow \alpha c \beta \mid \beta d \alpha$$

$$\alpha \rightarrow a \alpha \mid b b \alpha \mid \varepsilon$$

$$\beta \rightarrow a a \beta \mid \beta b \mid \varepsilon$$

Konštrukcia syntaktického stromu k slovu *aabbacaab*:



Syntaktická analýza zhora nadol (príklad)

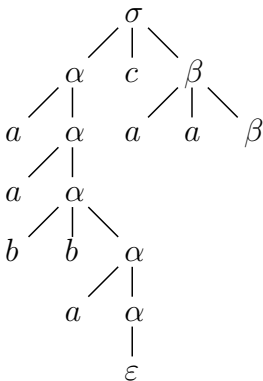
Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow \alpha c \beta \mid \beta d \alpha$$

$$\alpha \rightarrow a \alpha \mid b b \alpha \mid \varepsilon$$

$$\beta \rightarrow a a \beta \mid \beta b \mid \varepsilon$$

Konštrukcia syntaktického stromu k slovu *aabbacaab*:



Syntaktická analýza zhora nadol (príklad)

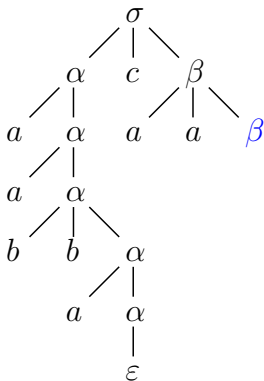
Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow \alpha c \beta \mid \beta d \alpha$$

$$\alpha \rightarrow a \alpha \mid b b \alpha \mid \varepsilon$$

$$\beta \rightarrow a a \beta \mid \beta b \mid \varepsilon$$

Konštrukcia syntaktického stromu k slovu *aabbacaab*:



Syntaktická analýza zhora nadol (príklad)

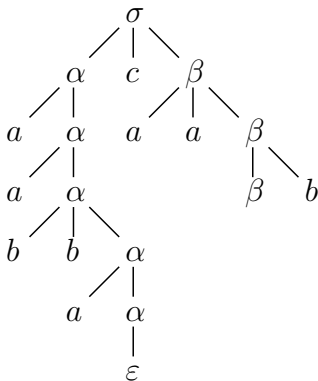
Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow \alpha c \beta \mid \beta d \alpha$$

$$\alpha \rightarrow a \alpha \mid b b \alpha \mid \varepsilon$$

$$\beta \rightarrow a a \beta \mid \beta b \mid \varepsilon$$

Konštrukcia syntaktického stromu k slovu *aabbacaab*:



Syntaktická analýza zhora nadol (príklad)

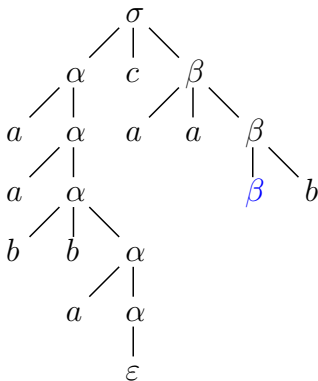
Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow \alpha c \beta \mid \beta d \alpha$$

$$\alpha \rightarrow a \alpha \mid b b \alpha \mid \varepsilon$$

$$\beta \rightarrow a a \beta \mid \beta b \mid \varepsilon$$

Konštrukcia syntaktického stromu k slovu *aabbacaab*:



Syntaktická analýza zhora nadol (príklad)

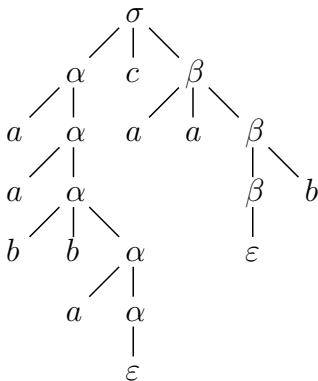
Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow \alpha c \beta \mid \beta d \alpha$$

$$\alpha \rightarrow a \alpha \mid b b \alpha \mid \varepsilon$$

$$\beta \rightarrow a a \beta \mid \beta b \mid \varepsilon$$

Konštrukcia syntaktického stromu k slovu *aabbacaab*:



Syntaktická analýza zhora nadol (príklad)

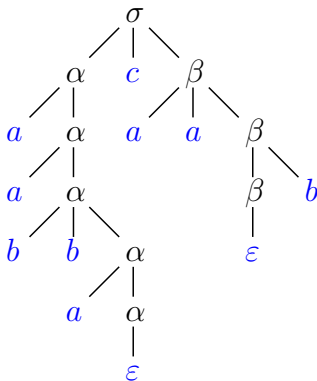
Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow \alpha c \beta \mid \beta d \alpha$$

$$\alpha \rightarrow a \alpha \mid b b \alpha \mid \varepsilon$$

$$\beta \rightarrow a a \beta \mid \beta b \mid \varepsilon$$

Konštrukcia syntaktického stromu k slovu *aabbacaab*:



Syntaktická analýza zhora nadol

- Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**

Syntaktická analýza zhora nadol

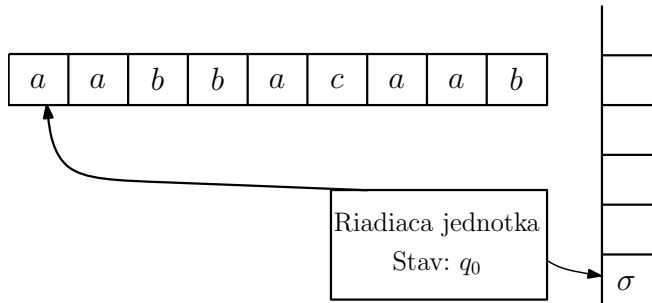
- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate

Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike

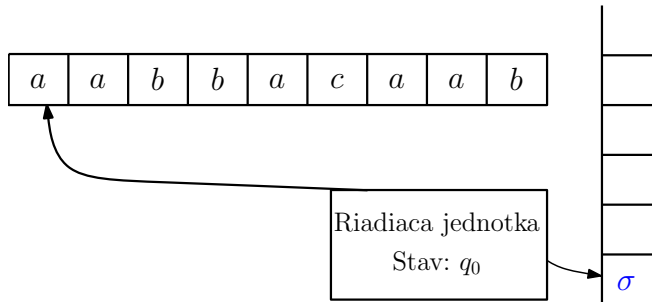
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



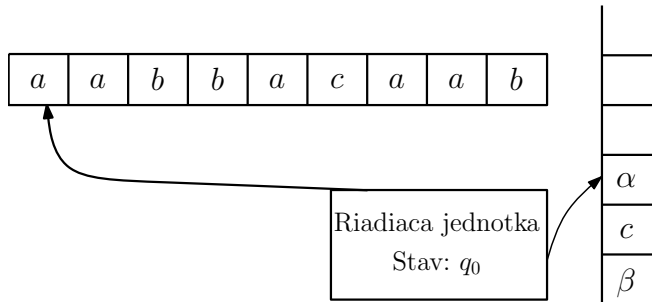
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



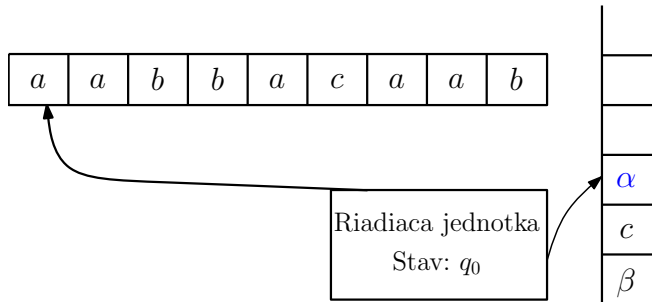
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



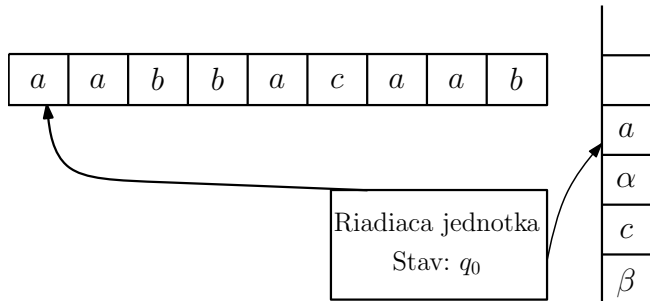
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



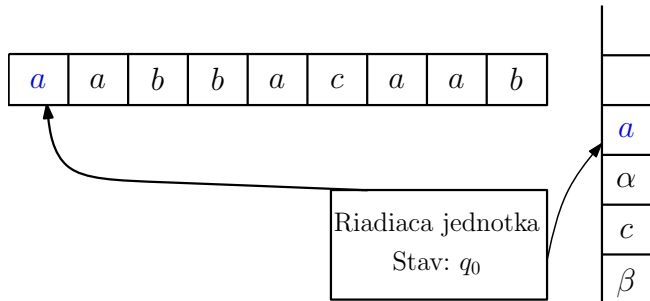
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



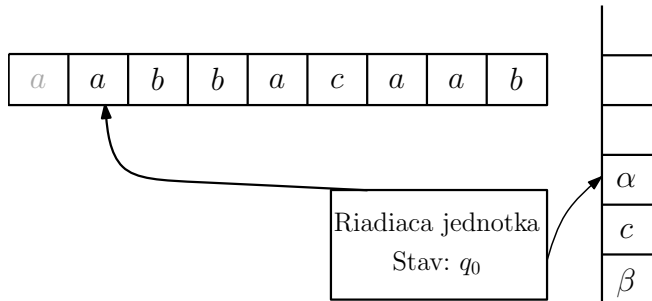
Syntaktická analýza zhora nadol

- Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



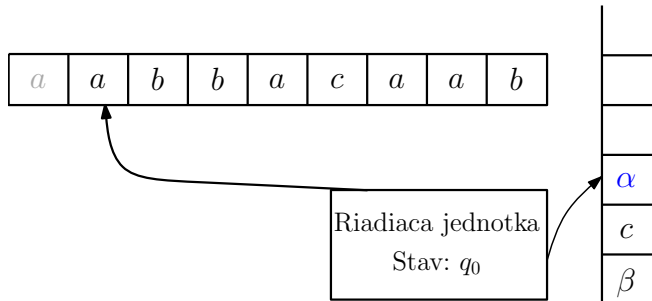
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



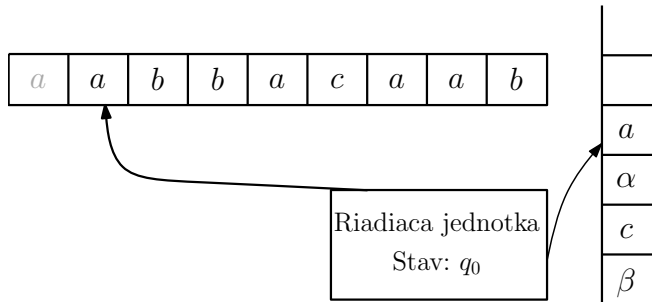
Syntaktická analýza zhora nadol

- Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



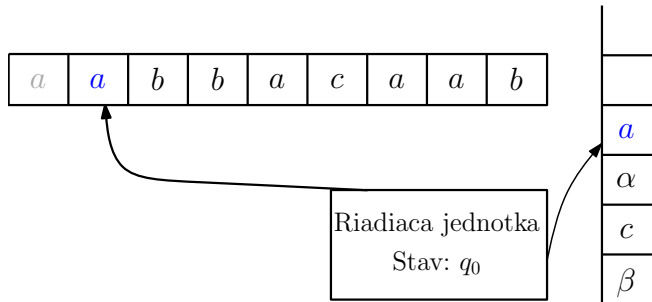
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



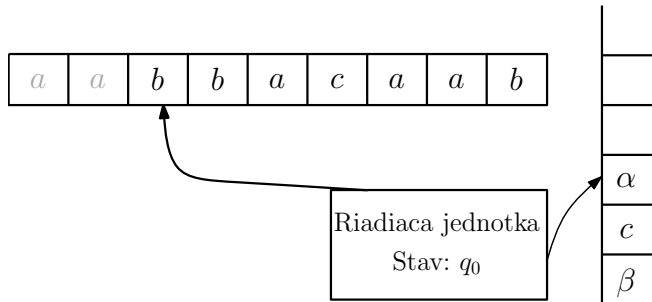
Syntaktická analýza zhora nadol

- Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



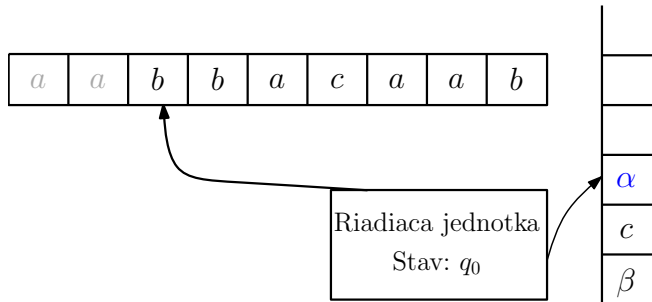
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



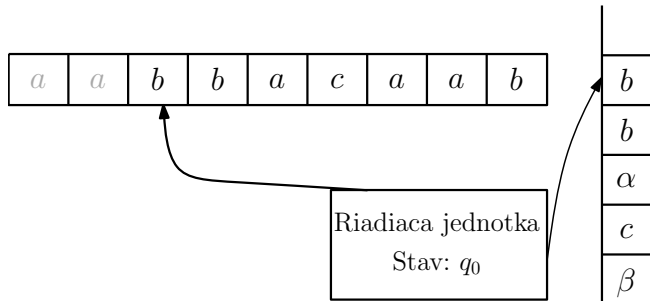
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



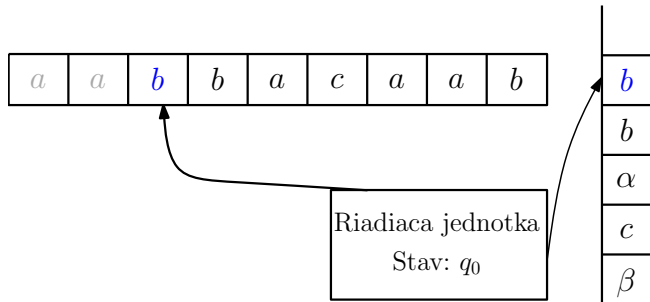
Syntaktická analýza zhora nadol

- Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



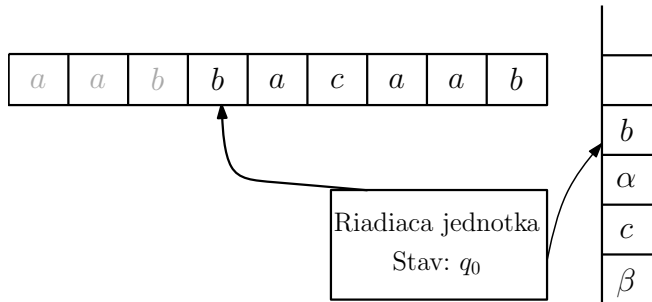
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



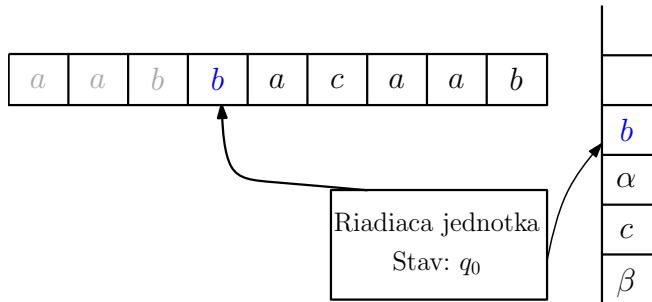
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



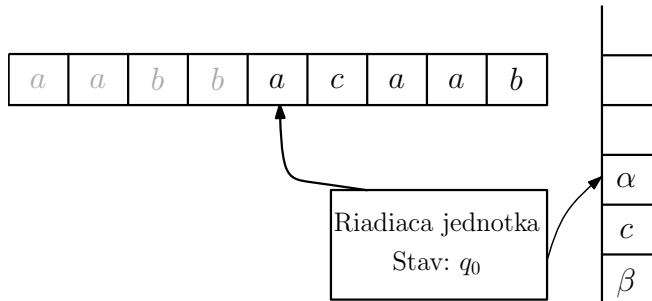
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



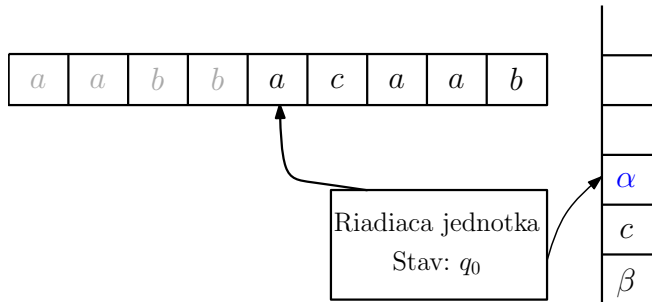
Syntaktická analýza zhora nadol

- Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



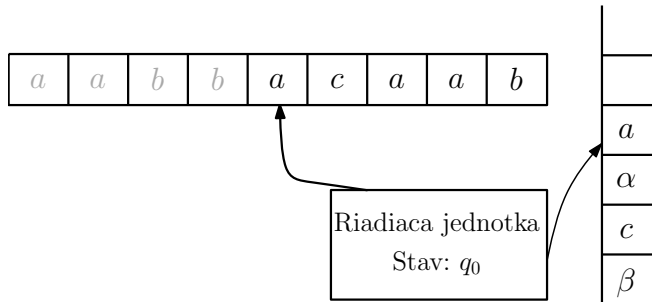
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



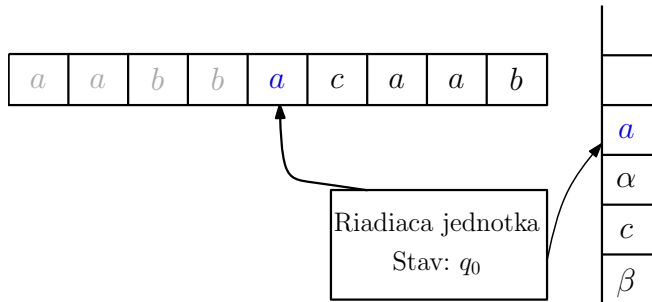
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



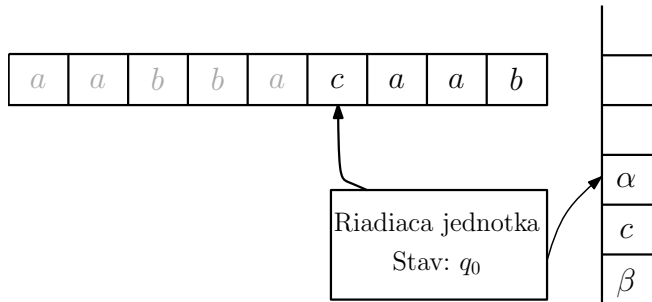
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



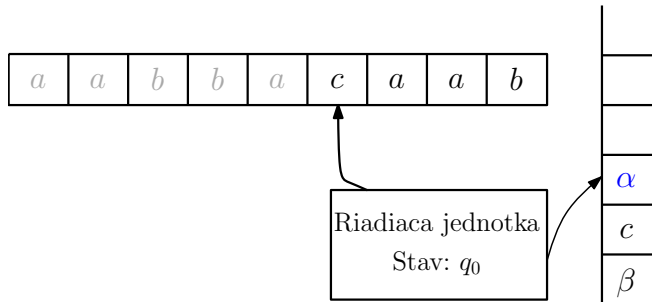
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



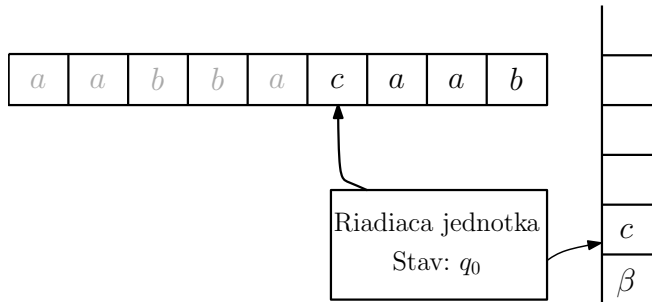
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



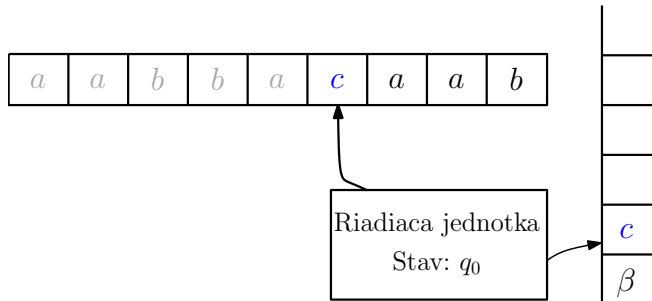
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



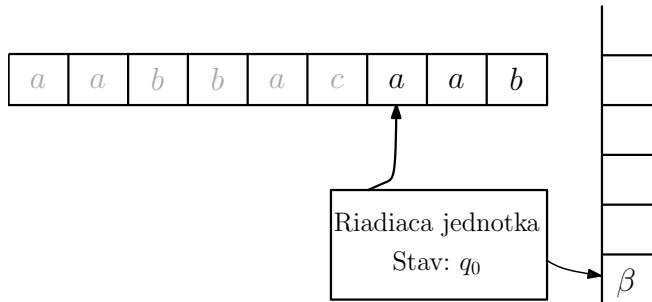
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



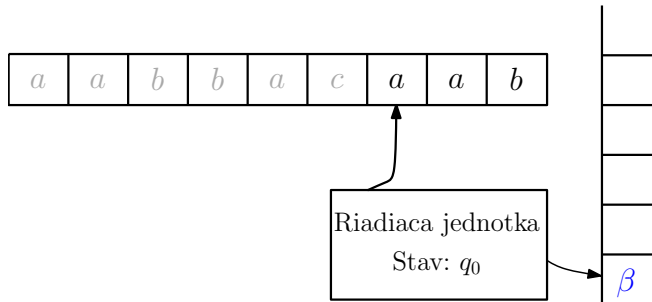
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



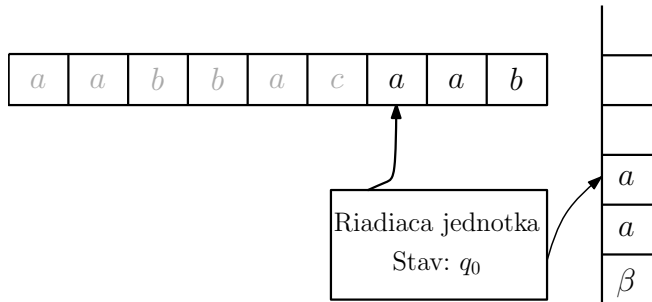
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



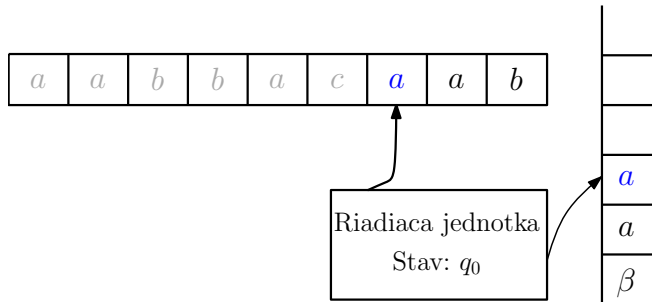
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



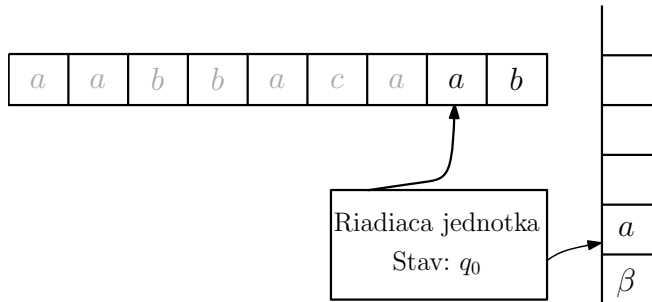
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



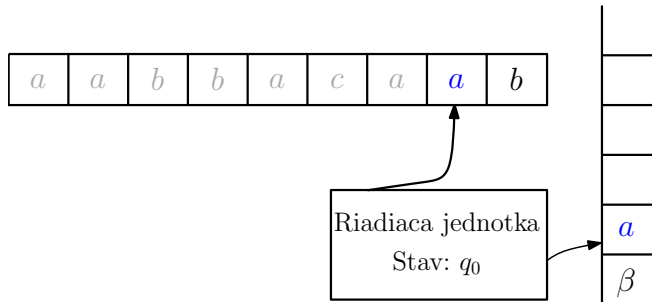
Syntaktická analýza zhora nadol

- Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



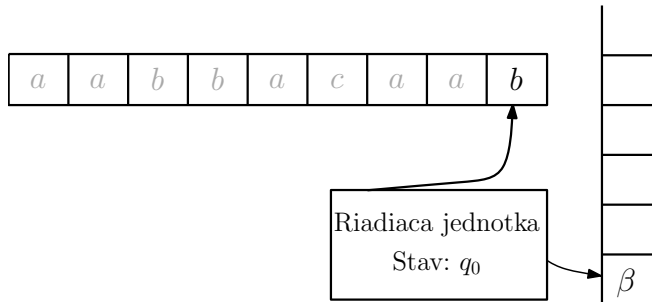
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



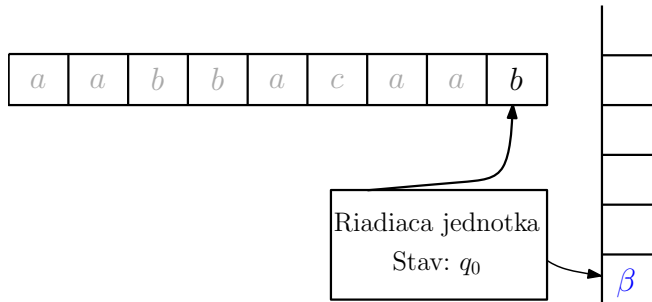
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



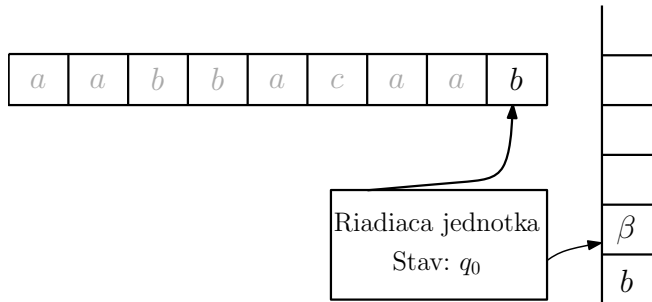
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



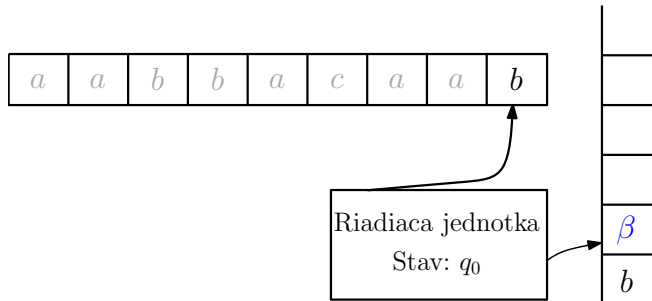
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



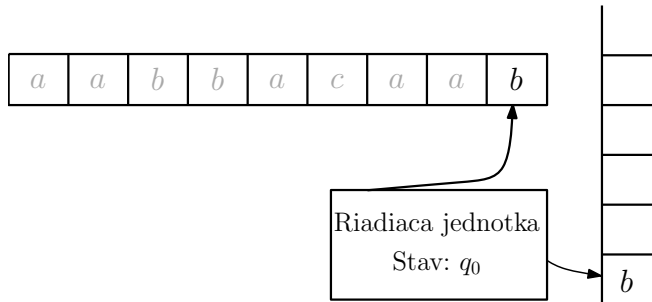
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



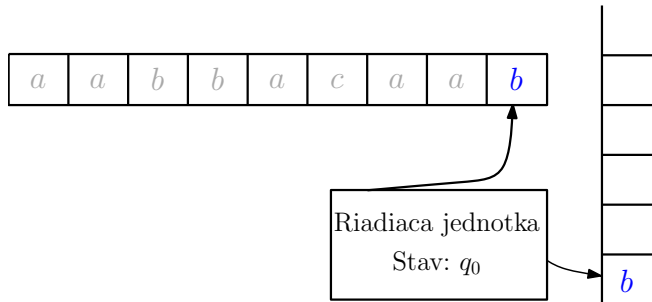
Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



Syntaktická analýza zhora nadol

- ▶ Doposiaľ nejde o algoritmus, ale len o **nedeterministickú schému algoritmu**
- ▶ Dá sa implementovať na (nedeterministickom) zásobníkovom automate
- ▶ Štandardná konštrukcia automatu k **ľubovoľnej** bezkontextovej gramatike



Metóda rekurzívneho zostupu

- ▶ Uvedenú nedeterministickú schému je potrebné determinizovať

Metóda rekurzívneho zostupu

- ▶ Uvedenú nedeterministickú schému je potrebné determinizovať

Metóda rekurzívneho zostupu:

Metóda rekurzívneho zostupu

- ▶ Uvedenú nedeterministickú schému je potrebné determinizovať

Metóda rekurzívneho zostupu:

- ▶ Úplné prehľadávanie cez všetky možné voľby pravidiel

Metóda rekurzívneho zostupu

- ▶ Uvedenú nedeterministickú schému je potrebné determinizovať

Metóda rekurzívneho zostupu:

- ▶ Úplné prehľadávanie cez všetky možné voľby pravidiel
- ▶ Exponenciálna časová zložitosť

Metóda rekurzívneho zostupu

- ▶ Uvedenú nedeterministickú schému je potrebné determinizovať

Metóda rekurzívneho zostupu:

- ▶ Úplné prehľadávanie cez všetky možné voľby pravidiel
- ▶ Exponenciálna časová zložitosť
- ▶ Ľavá rekúzia ($\alpha \Rightarrow^+ \alpha w$ pre nejaký neterminál α a slovo w) môže spôsobiť nekonečný cyklus

Metóda rekurzívneho zostupu

- ▶ Uvedenú nedeterministickú schému je potrebné determinizovať

Metóda rekurzívneho zostupu:

- ▶ Úplné prehľadávanie cez všetky možné voľby pravidiel
- ▶ Exponenciálna časová zložitosť
- ▶ Ľavá rekurzia ($\alpha \Rightarrow^+ \alpha w$ pre nejaký neterminál α a slovo w) môže spôsobiť nekonečný cyklus
- ▶ Pre gramatiky bez ľavej rekurzie túto metódu možno použiť, ale je neefektívna

Prediktívne parsovanie

- ▶ Trieda determinizácií, pre ktoré výsledný algoritmus (zvyčajne) pracuje v lineárnom čase

Prediktívne parsovanie

- ▶ Trieda determinizácií, pre ktoré výsledný algoritmus (zvyčajne) pracuje v lineárnom čase
- ▶ Rekurzívny zostup bez potreby úplného prehľadávania

Prediktívne parsovanie

- ▶ Trieda determinizácií, pre ktoré výsledný algoritmus (zvyčajne) pracuje v lineárnom čase
- ▶ Rekurzívny zostup bez potreby úplného prehľadávania
- ▶ Iba pre určité obmedzené (často ale postačujúce) triedy gramatík

Prediktívne parsovanie

- ▶ Trieda determinizácií, pre ktoré výsledný algoritmus (zvyčajne) pracuje v lineárnom čase
- ▶ Rekurzívny zostup bez potreby úplného prehľadávania
- ▶ Iba pre určité obmedzené (často ale postačujúce) triedy gramatík
- ▶ **Základná myšlienka:** pre každý neterminál ξ by sa na základe nejakej informácie o zvyšku vstupu (napríklad prvých k písmen) malo dať jednoznačne vybrať nasledujúce pravidlo $\xi \rightarrow x$

Prediktívne parsovanie

- ▶ Trieda determinizácií, pre ktoré výsledný algoritmus (zvyčajne) pracuje v lineárnom čase
- ▶ Rekurzívny zostup bez potreby úplného prehľadávania
- ▶ Iba pre určité obmedzené (často ale postačujúce) triedy gramatík
- ▶ **Základná myšlienka:** pre každý neterminál ξ by sa na základe nejakej informácie o zvyšku vstupu (napríklad prvých k písmen) malo dať jednoznačne vybrať nasledujúce pravidlo $\xi \rightarrow x$
- ▶ Napríklad SLL(k) gramatiky alebo LL(k) gramatiky

Časť 2:

SLL(k) a LL(k) gramatiky

Myšlienka $SLL(k)$ a $LL(k)$ gramatík

Pri $SLL(k)$ gramatikách je nasledujúce pravidlo jednoznačne určené:

Myšlienka $SLL(k)$ a $LL(k)$ gramatík

Pri $SLL(k)$ gramatikách je nasledujúce pravidlo jednoznačne určené:

- ▶ Neterminálom, ktorý sa má prepísať

Myšlienka $SLL(k)$ a $LL(k)$ gramatík

Pri $SLL(k)$ gramatikách je nasledujúce pravidlo jednoznačne určené:

- ▶ Neterminálom, ktorý sa má prepísať
- ▶ Nasledujúcimi k neprečítanými symbolmi na vstupe

Myšlienka $SLL(k)$ a $LL(k)$ gramatík

Pri $SLL(k)$ gramatikách je nasledujúce pravidlo jednoznačne určené:

- ▶ Neterminálom, ktorý sa má prepísať
- ▶ Nasledujúcimi k neprečítanými symbolmi na vstupe

Pri $LL(k)$ gramatikách je nasledujúce pravidlo jednoznačne určené:

Myšlienka $SLL(k)$ a $LL(k)$ gramatík

Pri $SLL(k)$ gramatikách je nasledujúce pravidlo jednoznačne určené:

- ▶ Neterminálom, ktorý sa má prepísať
- ▶ Nasledujúcimi k neprečítanými symbolmi na vstupe

Pri $LL(k)$ gramatikách je nasledujúce pravidlo jednoznačne určené:

- ▶ Neterminálom, ktorý sa má prepísať

Myšlienka $SLL(k)$ a $LL(k)$ gramatík

Pri $SLL(k)$ gramatikách je nasledujúce pravidlo jednoznačne určené:

- ▶ Neterminálom, ktorý sa má prepísať
- ▶ Nasledujúcimi k neprečítanými symbolmi na vstupe

Pri $LL(k)$ gramatikách je nasledujúce pravidlo jednoznačne určené:

- ▶ Neterminálom, ktorý sa má prepísať
- ▶ Nasledujúcimi k neprečítanými symbolmi na vstupe

Myšlienka SLL(k) a LL(k) gramatík

Pri SLL(k) gramatikách je nasledujúce pravidlo jednoznačne určené:

- ▶ Neterminálom, ktorý sa má prepísať
- ▶ Nasledujúcimi k neprečítanými symbolmi na vstupe

Pri LL(k) gramatikách je nasledujúce pravidlo jednoznačne určené:

- ▶ Neterminálom, ktorý sa má prepísať
- ▶ Nasledujúcimi k neprečítanými symbolmi na vstupe
- ▶ „Doterajším priebehom výpočtu parsovacieho algoritmu“

Myšlienka $SLL(k)$ a $LL(k)$ gramatík

Pri $SLL(k)$ gramatikách je nasledujúce pravidlo jednoznačne určené:

- ▶ Neterminálom, ktorý sa má prepísať
- ▶ Nasledujúcimi k neprečítanými symbolmi na vstupe

Pri $LL(k)$ gramatikách je nasledujúce pravidlo jednoznačne určené:

- ▶ Neterminálom, ktorý sa má prepísať
- ▶ Nasledujúcimi k neprečítanými symbolmi na vstupe
- ▶ „Doterajším priebehom výpočtu parsovacieho algoritmu“

Pôvod pomenovania $LL(k)$ resp. $SLL(k)$:

Myšlienka $SLL(k)$ a $LL(k)$ gramatík

Pri $SLL(k)$ gramatikách je nasledujúce pravidlo jednoznačne určené:

- ▶ Neterminálom, ktorý sa má prepísať
- ▶ Nasledujúcimi k neprečítanými symbolmi na vstupe

Pri $LL(k)$ gramatikách je nasledujúce pravidlo jednoznačne určené:

- ▶ Neterminálom, ktorý sa má prepísať
- ▶ Nasledujúcimi k neprečítanými symbolmi na vstupe
- ▶ „Doterajším priebehom výpočtu parsovacieho algoritmu“

Pôvod pomenovania $LL(k)$ resp. $SLL(k)$:

- ▶ $LL(k)$: **L**eft-to-right, **L**eftmost derivation, **k**-lookahead

Myšlienka $SLL(k)$ a $LL(k)$ gramatík

Pri $SLL(k)$ gramatikách je nasledujúce pravidlo jednoznačne určené:

- ▶ Neterminálom, ktorý sa má prepísať
- ▶ Nasledujúcimi k neprečítanými symbolmi na vstupe

Pri $LL(k)$ gramatikách je nasledujúce pravidlo jednoznačne určené:

- ▶ Neterminálom, ktorý sa má prepísať
- ▶ Nasledujúcimi k neprečítanými symbolmi na vstupe
- ▶ „Doterajším priebehom výpočtu parsovacieho algoritmu“

Pôvod pomenovania $LL(k)$ resp. $SLL(k)$:

- ▶ $LL(k)$: **L**eft-to-right, **L**eftmost derivation, **k**-lookahead
- ▶ $SLL(k)$: **S**trong- $LL(k)$; silnejšia je podmienka, nie gramatika

Množiny $\text{FIRST}_k(x)$ a $\text{FOLLOW}_k(x)$

- Nech $w = a_1 a_2 \dots a_n$, kde $a_1, \dots, a_n$ sú symboly

Množiny $\text{FIRST}_k(x)$ a $\text{FOLLOW}_k(x)$

- ▶ Nech $w = a_1 a_2 \dots a_n$, kde $a_1, \dots, a_n$ sú symboly
- ▶ Nech k je prirodzené číslo

Množiny $\text{FIRST}_k(x)$ a $\text{FOLLOW}_k(x)$

- ▶ Nech $w = a_1 a_2 \dots a_n$, kde $a_1, \dots, a_n$ sú symboly
- ▶ Nech k je prirodzené číslo

$$k:w = \begin{cases} w & \text{ak } |w| \leq k \\ a_1 \dots a_k & \text{ak } |w| > k \end{cases}$$

Množiny $\text{FIRST}_k(x)$ a $\text{FOLLOW}_k(x)$

- ▶ Nech $w = a_1 a_2 \dots a_n$, kde $a_1, \dots, a_n$ sú symboly
- ▶ Nech k je prirodzené číslo

$$k:w = \begin{cases} w & \text{ak } |w| \leq k \\ a_1 \dots a_k & \text{ak } |w| > k \end{cases}$$

Nech $G = (N, T, P, \sigma)$ je gramatika a $x \in (N \cup T)^*$ je slovo:

Množiny $\text{FIRST}_k(x)$ a $\text{FOLLOW}_k(x)$

- ▶ Nech $w = a_1 a_2 \dots a_n$, kde $a_1, \dots, a_n$ sú symboly
- ▶ Nech k je prirodzené číslo

$$k:w = \begin{cases} w & \text{ak } |w| \leq k \\ a_1 \dots a_k & \text{ak } |w| > k \end{cases}$$

Nech $G = (N, T, P, \sigma)$ je gramatika a $x \in (N \cup T)^*$ je slovo:

- ▶ $\text{FIRST}_k^G(x) = \{k:w \mid w \in T^*; x \Rightarrow^* w\}$

Množiny $\text{FIRST}_k(x)$ a $\text{FOLLOW}_k(x)$

- ▶ Nech $w = a_1 a_2 \dots a_n$, kde $a_1, \dots, a_n$ sú symboly
- ▶ Nech k je prirodzené číslo

$$k:w = \begin{cases} w & \text{ak } |w| \leq k \\ a_1 \dots a_k & \text{ak } |w| > k \end{cases}$$

Nech $G = (N, T, P, \sigma)$ je gramatika a $x \in (N \cup T)^*$ je slovo:

- ▶ $\text{FIRST}_k^G(x) = \{k:w \mid w \in T^*; x \Rightarrow^* w\}$
- ▶ $\text{FOLLOW}_k^G(x) = \bigcup_{\substack{w \in T^*, z \in (N \cup T)^* \\ \sigma \Rightarrow_{lm}^* wxz}} \text{FIRST}_k^G(z)$

Množiny $\text{FIRST}_k(x)$ a $\text{FOLLOW}_k(x)$

- ▶ Nech $w = a_1 a_2 \dots a_n$, kde $a_1, \dots, a_n$ sú symboly
- ▶ Nech k je prirodzené číslo

$$k:w = \begin{cases} w & \text{ak } |w| \leq k \\ a_1 \dots a_k & \text{ak } |w| > k \end{cases}$$

Nech $G = (N, T, P, \sigma)$ je gramatika a $x \in (N \cup T)^*$ je slovo:

- ▶ $\text{FIRST}_k^G(x) = \{k:w \mid w \in T^*; x \Rightarrow^* w\}$
- ▶ $\text{FOLLOW}_k^G(x) = \bigcup_{\substack{w \in T^*, z \in (N \cup T)^* \\ \sigma \Rightarrow_{lm}^* wxz}} \text{FIRST}_k^G(z)$
- ▶ Zvyčajne píšeme iba $\text{FIRST}_k(x)$ resp. $\text{FOLLOW}_k(x)$

Množiny $\text{FIRST}_k(x)$ a $\text{FOLLOW}_k(x)$

- ▶ Nech $w = a_1 a_2 \dots a_n$, kde $a_1, \dots, a_n$ sú symboly
- ▶ Nech k je prirodzené číslo

$$k:w = \begin{cases} w & \text{ak } |w| \leq k \\ a_1 \dots a_k & \text{ak } |w| > k \end{cases}$$

Nech $G = (N, T, P, \sigma)$ je gramatika a $x \in (N \cup T)^*$ je slovo:

- ▶ $\text{FIRST}_k^G(x) = \{k:w \mid w \in T^*; x \Rightarrow^* w\}$
- ▶ $\text{FOLLOW}_k^G(x) = \bigcup_{\substack{w \in T^*, z \in (N \cup T)^* \\ \sigma \Rightarrow_{lm}^* wxz}} \text{FIRST}_k^G(z)$
- ▶ Zvyčajne píšeme iba $\text{FIRST}_k(x)$ resp. $\text{FOLLOW}_k(x)$
- ▶ Prirodzené rozšírenie na $\text{FIRST}_k(L)$ resp. $\text{FOLLOW}_k(L)$ pre $L \subseteq (N \cup T)^*$

Množiny $\text{FIRST}_k(x)$ a $\text{FOLLOW}_k(x)$

- ▶ Nech $w = a_1 a_2 \dots a_n$, kde $a_1, \dots, a_n$ sú symboly
- ▶ Nech k je prirodzené číslo

$$k:w = \begin{cases} w & \text{ak } |w| \leq k \\ a_1 \dots a_k & \text{ak } |w| > k \end{cases}$$

Nech $G = (N, T, P, \sigma)$ je gramatika a $x \in (N \cup T)^*$ je slovo:

- ▶ $\text{FIRST}_k^G(x) = \{k:w \mid w \in T^*; x \Rightarrow^* w\}$
- ▶ $\text{FOLLOW}_k^G(x) = \bigcup_{\substack{w \in T^*, z \in (N \cup T)^* \\ \sigma \Rightarrow_{lm}^* wxz}} \text{FIRST}_k^G(z)$
- ▶ Zvyčajne píšeme iba $\text{FIRST}_k(x)$ resp. $\text{FOLLOW}_k(x)$
- ▶ Prirodzené rozšírenie na $\text{FIRST}_k(L)$ resp. $\text{FOLLOW}_k(L)$ pre $L \subseteq (N \cup T)^*$
- ▶ Trochu iná notácia ako pri jednoducho precedenčných gramatikách (Formálne jazyky a automaty 2)

SLL(k) a LL(k) gramatiky

Definícia

SLL(k) gramatika je bezkontextová gramatika $G = (N, T, P, \sigma)$ taká, že

SLL(k) a LL(k) gramatiky

Definícia

SLL(k) gramatika je bezkontextová gramatika $G = (N, T, P, \sigma)$ taká, že

- ▶ pre všetky neterminály $\xi \in N$

SLL(k) a LL(k) gramatiky

Definícia

SLL(k) gramatika je bezkontextová gramatika $G = (N, T, P, \sigma)$ taká, že

- ▶ pre všetky neterminály $\xi \in N$
- ▶ a všetky dvojice pravidiel $\xi \rightarrow x_1, \xi \rightarrow x_2 \in P$, kde $x_1 \neq x_2$

SLL(k) a LL(k) gramatiky

Definícia

SLL(k) gramatika je bezkontextová gramatika $G = (N, T, P, \sigma)$ taká, že

- ▶ pre všetky neterminály $\xi \in N$
- ▶ a všetky dvojice pravidiel $\xi \rightarrow x_1, \xi \rightarrow x_2 \in P$, kde $x_1 \neq x_2$

platí $\text{FIRST}_k(x_1\text{FOLLOW}_k(\xi)) \cap \text{FIRST}_k(x_2\text{FOLLOW}_k(\xi)) = \emptyset$.

SLL(k) a LL(k) gramatiky

Definícia

SLL(k) gramatika je bezkontextová gramatika $G = (N, T, P, \sigma)$ taká, že

- ▶ pre všetky neterminály $\xi \in N$
- ▶ a všetky dvojice pravidiel $\xi \rightarrow x_1, \xi \rightarrow x_2 \in P$, kde $x_1 \neq x_2$

platí $\text{FIRST}_k(x_1\text{FOLLOW}_k(\xi)) \cap \text{FIRST}_k(x_2\text{FOLLOW}_k(\xi)) = \emptyset$.

Definícia

LL(k) gramatika je bezkontextová gramatika $G = (N, T, P, \sigma)$ taká, že

SLL(k) a LL(k) gramatiky

Definícia

SLL(k) gramatika je bezkontextová gramatika $G = (N, T, P, \sigma)$ taká, že

- ▶ pre všetky neterminály $\xi \in N$
- ▶ a všetky dvojice pravidiel $\xi \rightarrow x_1, \xi \rightarrow x_2 \in P$, kde $x_1 \neq x_2$

platí $\text{FIRST}_k(x_1\text{FOLLOW}_k(\xi)) \cap \text{FIRST}_k(x_2\text{FOLLOW}_k(\xi)) = \emptyset$.

Definícia

LL(k) gramatika je bezkontextová gramatika $G = (N, T, P, \sigma)$ taká, že

- ▶ pre všetky neterminály $\xi \in N$,

SLL(k) a LL(k) gramatiky

Definícia

SLL(k) gramatika je bezkontextová gramatika $G = (N, T, P, \sigma)$ taká, že

- ▶ pre všetky neterminály $\xi \in N$
- ▶ a všetky dvojice pravidiel $\xi \rightarrow x_1, \xi \rightarrow x_2 \in P$, kde $x_1 \neq x_2$

platí $\text{FIRST}_k(x_1\text{FOLLOW}_k(\xi)) \cap \text{FIRST}_k(x_2\text{FOLLOW}_k(\xi)) = \emptyset$.

Definícia

LL(k) gramatika je bezkontextová gramatika $G = (N, T, P, \sigma)$ taká, že

- ▶ pre všetky neterminály $\xi \in N$,
- ▶ všetky dvojice pravidiel $\xi \rightarrow x_1, \xi \rightarrow x_2 \in P$, kde $x_1 \neq x_2$

SLL(k) a LL(k) gramatiky

Definícia

SLL(k) gramatika je bezkontextová gramatika $G = (N, T, P, \sigma)$ taká, že

- ▶ pre všetky neterminály $\xi \in N$
- ▶ a všetky dvojice pravidiel $\xi \rightarrow x_1, \xi \rightarrow x_2 \in P$, kde $x_1 \neq x_2$

platí $\text{FIRST}_k(x_1 \text{ FOLLOW}_k(\xi)) \cap \text{FIRST}_k(x_2 \text{ FOLLOW}_k(\xi)) = \emptyset$.

Definícia

LL(k) gramatika je bezkontextová gramatika $G = (N, T, P, \sigma)$ taká, že

- ▶ pre všetky neterminály $\xi \in N$,
- ▶ všetky dvojice pravidiel $\xi \rightarrow x_1, \xi \rightarrow x_2 \in P$, kde $x_1 \neq x_2$
- ▶ a všetky odvodenia $\sigma \Rightarrow_{lm}^* w\xi z$, kde $w \in T^*$ a $z \in (N \cup T)^*$

SLL(k) a LL(k) gramatiky

Definícia

SLL(k) gramatika je bezkontextová gramatika $G = (N, T, P, \sigma)$ taká, že

- ▶ pre všetky neterminály $\xi \in N$
- ▶ a všetky dvojice pravidiel $\xi \rightarrow x_1, \xi \rightarrow x_2 \in P$, kde $x_1 \neq x_2$

platí $\text{FIRST}_k(x_1 \text{ FOLLOW}_k(\xi)) \cap \text{FIRST}_k(x_2 \text{ FOLLOW}_k(\xi)) = \emptyset$.

Definícia

LL(k) gramatika je bezkontextová gramatika $G = (N, T, P, \sigma)$ taká, že

- ▶ pre všetky neterminály $\xi \in N$,
- ▶ všetky dvojice pravidiel $\xi \rightarrow x_1, \xi \rightarrow x_2 \in P$, kde $x_1 \neq x_2$
- ▶ a všetky odvodenia $\sigma \Rightarrow_{lm}^* w\xi z$, kde $w \in T^*$ a $z \in (N \cup T)^*$

platí $\text{FIRST}_k(x_1 z) \cap \text{FIRST}_k(x_2 z) = \emptyset$.

SLL(k) a LL(k) gramatiky

Definícia

SLL(k) gramatika je bezkontextová gramatika $G = (N, T, P, \sigma)$ taká, že

- ▶ pre všetky neterminály $\xi \in N$
- ▶ a všetky dvojice pravidiel $\xi \rightarrow x_1, \xi \rightarrow x_2 \in P$, kde $x_1 \neq x_2$

platí $\text{FIRST}_k(x_1 \text{ FOLLOW}_k(\xi)) \cap \text{FIRST}_k(x_2 \text{ FOLLOW}_k(\xi)) = \emptyset$.

Definícia

LL(k) gramatika je bezkontextová gramatika $G = (N, T, P, \sigma)$ taká, že

- ▶ pre všetky neterminály $\xi \in N$,
- ▶ všetky dvojice pravidiel $\xi \rightarrow x_1, \xi \rightarrow x_2 \in P$, kde $x_1 \neq x_2$
- ▶ a všetky odvodenia $\sigma \Rightarrow_{lm}^* w\xi z$, kde $w \in T^*$ a $z \in (N \cup T)^*$

platí $\text{FIRST}_k(x_1 z) \cap \text{FIRST}_k(x_2 z) = \emptyset$.

- ▶ Každá SLL(k) gramatika je očividne aj LL(k) gramatika

Príklad SLL(1) gramatiky

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

Príklad SLL(1) gramatiky

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow aa \mid b\sigma\alpha\alpha$$

$$\alpha \rightarrow a\alpha\sigma \mid b$$

Príklad SLL(1) gramatiky

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow aa \mid b\sigma\alpha\alpha$$

$$\alpha \rightarrow a\alpha\sigma \mid b$$

Zrejme platí:

Príklad SLL(1) gramatiky

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow aa \mid b\sigma\alpha\alpha$$

$$\alpha \rightarrow a\alpha\sigma \mid b$$

Zrejme platí:

- ▶ $\text{FIRST}_1(aa\text{FOLLOW}_1(\sigma)) = \{a\}$

Príklad SLL(1) gramatiky

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow aa \mid b\sigma\alpha\alpha$$

$$\alpha \rightarrow a\alpha\sigma \mid b$$

Zrejme platí:

- ▶ $\text{FIRST}_1(aa\text{FOLLOW}_1(\sigma)) = \{a\}$
- ▶ $\text{FIRST}_1(b\sigma\alpha\alpha\text{FOLLOW}_1(\sigma)) = \{b\}$

Príklad SLL(1) gramatiky

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow aa \mid b\sigma\alpha\alpha$$

$$\alpha \rightarrow a\alpha\sigma \mid b$$

Zrejme platí:

- ▶ $\text{FIRST}_1(aa\text{FOLLOW}_1(\sigma)) = \{a\}$
- ▶ $\text{FIRST}_1(b\sigma\alpha\alpha\text{FOLLOW}_1(\sigma)) = \{b\}$
- ▶ $\{a\} \cap \{b\} = \emptyset$

Príklad SLL(1) gramatiky

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow aa \mid b\sigma\alpha\alpha$$

$$\alpha \rightarrow a\alpha\sigma \mid b$$

Zrejme platí:

- ▶ $\text{FIRST}_1(aa\text{FOLLOW}_1(\sigma)) = \{a\}$
- ▶ $\text{FIRST}_1(b\sigma\alpha\alpha\text{FOLLOW}_1(\sigma)) = \{b\}$
- ▶ $\{a\} \cap \{b\} = \emptyset$
- ▶ $\text{FIRST}_1(a\alpha\sigma\text{FOLLOW}_1(\alpha)) = \{a\}$

Príklad SLL(1) gramatiky

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow aa \mid b\sigma\alpha\alpha$$

$$\alpha \rightarrow a\alpha\sigma \mid b$$

Zrejme platí:

- ▶ $\text{FIRST}_1(aa\text{FOLLOW}_1(\sigma)) = \{a\}$
- ▶ $\text{FIRST}_1(b\sigma\alpha\alpha\text{FOLLOW}_1(\sigma)) = \{b\}$
- ▶ $\{a\} \cap \{b\} = \emptyset$
- ▶ $\text{FIRST}_1(a\alpha\sigma\text{FOLLOW}_1(\alpha)) = \{a\}$
- ▶ $\text{FIRST}_1(b\text{FOLLOW}_1(\alpha)) = \{b\}$

Príklad SLL(1) gramatiky

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow aa \mid b\sigma\alpha\alpha$$

$$\alpha \rightarrow a\alpha\sigma \mid b$$

Zrejme platí:

- ▶ $\text{FIRST}_1(aa\text{FOLLOW}_1(\sigma)) = \{a\}$
- ▶ $\text{FIRST}_1(b\sigma\alpha\alpha\text{FOLLOW}_1(\sigma)) = \{b\}$
- ▶ $\{a\} \cap \{b\} = \emptyset$
- ▶ $\text{FIRST}_1(a\alpha\sigma\text{FOLLOW}_1(\alpha)) = \{a\}$
- ▶ $\text{FIRST}_1(b\text{FOLLOW}_1(\alpha)) = \{b\}$
- ▶ $\{a\} \cap \{b\} = \emptyset$

Príklad SLL(1) gramatiky

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow aa \mid b\sigma\alpha\alpha$$

$$\alpha \rightarrow a\alpha\sigma \mid b$$

Zrejme platí:

- ▶ $\text{FIRST}_1(aa\text{FOLLOW}_1(\sigma)) = \{a\}$
- ▶ $\text{FIRST}_1(b\sigma\alpha\alpha\text{FOLLOW}_1(\sigma)) = \{b\}$
- ▶ $\{a\} \cap \{b\} = \emptyset$
- ▶ $\text{FIRST}_1(a\alpha\sigma\text{FOLLOW}_1(\alpha)) = \{a\}$
- ▶ $\text{FIRST}_1(b\text{FOLLOW}_1(\alpha)) = \{b\}$
- ▶ $\{a\} \cap \{b\} = \emptyset$
- ▶ Gramatika je teda SLL(1)

Príklad SLL(1) gramatiky

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow aa \mid b\sigma\alpha\alpha$$

$$\alpha \rightarrow a\alpha\sigma \mid b$$

Zrejme platí:

- ▶ $\text{FIRST}_1(aa\text{FOLLOW}_1(\sigma)) = \{a\}$
- ▶ $\text{FIRST}_1(b\sigma\alpha\alpha\text{FOLLOW}_1(\sigma)) = \{b\}$
- ▶ $\{a\} \cap \{b\} = \emptyset$
- ▶ $\text{FIRST}_1(a\alpha\sigma\text{FOLLOW}_1(\alpha)) = \{a\}$
- ▶ $\text{FIRST}_1(b\text{FOLLOW}_1(\alpha)) = \{b\}$
- ▶ $\{a\} \cap \{b\} = \emptyset$
- ▶ Gramatika je teda SLL(1)
- ▶ Podobne pre všetky gramatiky také, že pre každý neterminál ξ začínajú prave strany pravidiel zo ξ po dvoch rôznymi terminálmi (ak zvyšok pravej strany pozostáva z neterminálov: [s-gramatiky](#))

Príklad gramatiky, ktorá nie je LL(1)

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

Príklad gramatiky, ktorá nie je LL(1)

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow aa\alpha \mid \varepsilon$$

$$\alpha \rightarrow \sigma ab \mid b$$

Príklad gramatiky, ktorá nie je LL(1)

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow aa\alpha \mid \varepsilon$$

$$\alpha \rightarrow \sigma ab \mid b$$

V tejto gramatike existuje napríklad nasledujúce odvodenie:

Príklad gramatiky, ktorá nie je LL(1)

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow aa\alpha \mid \varepsilon$$

$$\alpha \rightarrow \sigma ab \mid b$$

V tejto gramatike existuje napríklad nasledujúce odvodenie:

$$\sigma \Rightarrow_{lm} aa\alpha \Rightarrow_{lm} aa\sigma ab$$

Príklad gramatiky, ktorá nie je LL(1)

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow aa\alpha \mid \varepsilon$$

$$\alpha \rightarrow \sigma ab \mid b$$

V tejto gramatike existuje napríklad nasledujúce odvodenie:

$$\sigma \Rightarrow_{lm} aa\alpha \Rightarrow_{lm} aa\sigma ab$$

Platí:

Príklad gramatiky, ktorá nie je LL(1)

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow aa\alpha \mid \varepsilon$$

$$\alpha \rightarrow \sigma ab \mid b$$

V tejto gramatike existuje napríklad nasledujúce odvodenie:

$$\sigma \Rightarrow_{lm} aa\alpha \Rightarrow_{lm} aa\sigma ab$$

Platí:

$$\blacktriangleright \text{FIRST}_1(aa\alpha ab) = \{a\}$$

Príklad gramatiky, ktorá nie je LL(1)

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow aa\alpha \mid \varepsilon$$

$$\alpha \rightarrow \sigma ab \mid b$$

V tejto gramatike existuje napríklad nasledujúce odvodenie:

$$\sigma \Rightarrow_{lm} aa\alpha \Rightarrow_{lm} aa\sigma ab$$

Platí:

- ▶ $\text{FIRST}_1(aa\alpha ab) = \{a\}$
- ▶ $\text{FIRST}_1(\varepsilon ab) = \{a\}$

Príklad gramatiky, ktorá nie je LL(1)

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow aa\alpha \mid \varepsilon$$

$$\alpha \rightarrow \sigma ab \mid b$$

V tejto gramatike existuje napríklad nasledujúce odvodenie:

$$\sigma \Rightarrow_{lm} aa\alpha \Rightarrow_{lm} aa\sigma ab$$

Platí:

- ▶ $\text{FIRST}_1(aa\alpha ab) = \{a\}$
- ▶ $\text{FIRST}_1(\varepsilon ab) = \{a\}$
- ▶ $\{a\} \cap \{a\} \neq \emptyset$

Príklad gramatiky, ktorá nie je LL(1)

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow aa\alpha \mid \varepsilon$$

$$\alpha \rightarrow \sigma ab \mid b$$

V tejto gramatike existuje napríklad nasledujúce odvodenie:

$$\sigma \Rightarrow_{lm} aa\alpha \Rightarrow_{lm} aa\sigma ab$$

Platí:

- ▶ $\text{FIRST}_1(aa\alpha ab) = \{a\}$
- ▶ $\text{FIRST}_1(\varepsilon ab) = \{a\}$
- ▶ $\{a\} \cap \{a\} \neq \emptyset$
- ▶ Gramatika nie je LL(1)

Príklad gramatiky, ktorá nie je LL(1)

Uvažujme bezkontextovú gramatiku s nasledujúcimi pravidlami:

$$\sigma \rightarrow aa\alpha \mid \varepsilon$$

$$\alpha \rightarrow \sigma ab \mid b$$

V tejto gramatike existuje napríklad nasledujúce odvodenie:

$$\sigma \Rightarrow_{lm} aa\alpha \Rightarrow_{lm} aa\sigma ab$$

Platí:

- ▶ $\text{FIRST}_1(aa\alpha ab) = \{a\}$
- ▶ $\text{FIRST}_1(\varepsilon ab) = \{a\}$
- ▶ $\{a\} \cap \{a\} \neq \emptyset$
- ▶ Gramatika nie je LL(1)
- ▶ Gramatika je SLL(2), a teda aj LL(2) (cvičenie)

SLL(1) vs. LL(1) na úrovni gramatík

- Každá SLL(k) gramatika je očividne aj LL(k) gramatika

SLL(1) vs. LL(1) na úrovni gramatík

- ▶ Každá SLL(k) gramatika je očividne aj LL(k) gramatika
- ▶ Ukážeme, že každá LL(1) gramatika je súčasne aj SLL(1) gramatika

SLL(1) vs. LL(1) na úrovni gramatík

- ▶ Každá SLL(k) gramatika je očividne aj LL(k) gramatika
- ▶ Ukážeme, že každá LL(1) gramatika je súčasne aj SLL(1) gramatika
- ▶ Sporom, nech $G = (N, T, P, \sigma)$ je LL(1), ale nie je SLL(1)

SLL(1) vs. LL(1) na úrovni gramatík

- ▶ Každá SLL(k) gramatika je očividne aj LL(k) gramatika
- ▶ Ukážeme, že každá LL(1) gramatika je súčasne aj SLL(1) gramatika
- ▶ Sporom, nech $G = (N, T, P, \sigma)$ je LL(1), ale nie je SLL(1)

Keďže G nie je SLL(1):

SLL(1) vs. LL(1) na úrovni gramatík

- ▶ Každá SLL(k) gramatika je očividne aj LL(k) gramatika
- ▶ Ukážeme, že každá LL(1) gramatika je súčasne aj SLL(1) gramatika
- ▶ Sporom, nech $G = (N, T, P, \sigma)$ je LL(1), ale nie je SLL(1)

Keďže G nie je SLL(1):

- ▶ $\text{FIRST}_1(x_1\text{FOLLOW}_1(\xi)) \cap \text{FIRST}_1(x_2\text{FOLLOW}_1(\xi)) \neq \emptyset$
pre nejakú dvojicu rôznych pravidiel $\xi \rightarrow x_1, \xi \rightarrow x_2 \in P$

SLL(1) vs. LL(1) na úrovni gramatík

- ▶ Každá SLL(k) gramatika je očividne aj LL(k) gramatika
- ▶ Ukážeme, že každá LL(1) gramatika je súčasne aj SLL(1) gramatika
- ▶ Sporom, nech $G = (N, T, P, \sigma)$ je LL(1), ale nie je SLL(1)

Keďže G nie je SLL(1):

- ▶ $\text{FIRST}_1(x_1\text{FOLLOW}_1(\xi)) \cap \text{FIRST}_1(x_2\text{FOLLOW}_1(\xi)) \neq \emptyset$
pre nejakú dvojicu rôznych pravidiel $\xi \rightarrow x_1, \xi \rightarrow x_2 \in P$
- ▶ Nutne $\text{FOLLOW}_1(\xi) \neq \emptyset$; existuje teda aspoň 1 odvodenie $\sigma \Rightarrow_{lm}^* w\xi z$, kde $w \in T^*$ a $z \in (N \cup T)^*$ je „terminujúce“

SLL(1) vs. LL(1) na úrovni gramatík

- ▶ Každá SLL(k) gramatika je očividne aj LL(k) gramatika
- ▶ Ukážeme, že každá LL(1) gramatika je súčasne aj SLL(1) gramatika
- ▶ Sporom, nech $G = (N, T, P, \sigma)$ je LL(1), ale nie je SLL(1)

Keďže G nie je SLL(1):

- ▶ $\text{FIRST}_1(x_1\text{FOLLOW}_1(\xi)) \cap \text{FIRST}_1(x_2\text{FOLLOW}_1(\xi)) \neq \emptyset$
pre nejakú dvojicu rôznych pravidiel $\xi \rightarrow x_1, \xi \rightarrow x_2 \in P$
- ▶ Nutne $\text{FOLLOW}_1(\xi) \neq \emptyset$; existuje teda aspoň 1 ododenie $\sigma \Rightarrow_{im}^* w\xi z$, kde $w \in T^*$ a $z \in (N \cup T)^*$ je „terminujúce“
- ▶ Ak $x_1 \Rightarrow^* \varepsilon$ a $x_2 \Rightarrow^* \varepsilon$, tak $(\text{FIRST}_1(x_1 z) \cap \text{FIRST}_1(x_2 z)) \supseteq \text{FIRST}_1(z) \neq \emptyset$: spor

SLL(1) vs. LL(1) na úrovni gramatík

- ▶ Každá SLL(k) gramatika je očividne aj LL(k) gramatika
- ▶ Ukážeme, že každá LL(1) gramatika je súčasne aj SLL(1) gramatika
- ▶ Sporom, nech $G = (N, T, P, \sigma)$ je LL(1), ale nie je SLL(1)

Keďže G nie je SLL(1):

- ▶ $\text{FIRST}_1(x_1\text{FOLLOW}_1(\xi)) \cap \text{FIRST}_1(x_2\text{FOLLOW}_1(\xi)) \neq \emptyset$
pre nejakú dvojicu rôznych pravidiel $\xi \rightarrow x_1, \xi \rightarrow x_2 \in P$
- ▶ Nutne $\text{FOLLOW}_1(\xi) \neq \emptyset$; existuje teda aspoň 1 ododenie $\sigma \Rightarrow_{lm}^* w\xi z$, kde $w \in T^*$ a $z \in (N \cup T)^*$ je „terminujúce“
- ▶ Ak $x_1 \Rightarrow^* \varepsilon$ a $x_2 \Rightarrow^* \varepsilon$, tak $(\text{FIRST}_1(x_1 z) \cap \text{FIRST}_1(x_2 z)) \supseteq \text{FIRST}_1(z) \neq \emptyset$: spor
- ▶ Ak $x_1 \not\Rightarrow^* \varepsilon$ a $x_2 \not\Rightarrow^* \varepsilon$, tak $\text{FIRST}_1(x_1) \cap \text{FIRST}_1(x_2) \neq \emptyset$, a teda aj $\text{FIRST}_1(x_1 z) \cap \text{FIRST}_1(x_2 z) \neq \emptyset$: spor

SLL(1) vs. LL(1) na úrovni gramatík

- ▶ Každá SLL(k) gramatika je očividne aj LL(k) gramatika
- ▶ Ukážeme, že každá LL(1) gramatika je súčasne aj SLL(1) gramatika
- ▶ Sporom, nech $G = (N, T, P, \sigma)$ je LL(1), ale nie je SLL(1)

Keďže G nie je SLL(1):

- ▶ $\text{FIRST}_1(x_1 \text{ FOLLOW}_1(\xi)) \cap \text{FIRST}_1(x_2 \text{ FOLLOW}_1(\xi)) \neq \emptyset$
pre nejakú dvojicu rôznych pravidiel $\xi \rightarrow x_1, \xi \rightarrow x_2 \in P$
- ▶ Nutne $\text{FOLLOW}_1(\xi) \neq \emptyset$; existuje teda aspoň 1 ododenie $\sigma \Rightarrow_{im}^* w\xi z$, kde $w \in T^*$ a $z \in (N \cup T)^*$ je „terminujúce“
- ▶ Ak $x_1 \Rightarrow^* \varepsilon$ a $x_2 \Rightarrow^* \varepsilon$, tak $(\text{FIRST}_1(x_1 z) \cap \text{FIRST}_1(x_2 z)) \supseteq \text{FIRST}_1(z) \neq \emptyset$: spor
- ▶ Ak $x_1 \not\Rightarrow^* \varepsilon$ a $x_2 \not\Rightarrow^* \varepsilon$, tak $\text{FIRST}_1(x_1) \cap \text{FIRST}_1(x_2) \neq \emptyset$, a teda aj $\text{FIRST}_1(x_1 z) \cap \text{FIRST}_1(x_2 z) \neq \emptyset$: spor
- ▶ Ak $x_1 \not\Rightarrow^* \varepsilon$ a $x_2 \Rightarrow^* \varepsilon$, tak $\text{FIRST}_1(x_1) \cap \text{FIRST}_1(x_2) \neq \emptyset$ alebo $\text{FIRST}_1(x_1) \cap \text{FOLLOW}_1(\xi) \neq \emptyset$

SLL(1) vs. LL(1) na úrovni gramatík

- ▶ Každá SLL(k) gramatika je očividne aj LL(k) gramatika
- ▶ Ukážeme, že každá LL(1) gramatika je súčasne aj SLL(1) gramatika
- ▶ Sporom, nech $G = (N, T, P, \sigma)$ je LL(1), ale nie je SLL(1)

Keďže G nie je SLL(1):

- ▶ $\text{FIRST}_1(x_1\text{FOLLOW}_1(\xi)) \cap \text{FIRST}_1(x_2\text{FOLLOW}_1(\xi)) \neq \emptyset$
pre nejakú dvojicu rôznych pravidiel $\xi \rightarrow x_1, \xi \rightarrow x_2 \in P$
- ▶ Nutne $\text{FOLLOW}_1(\xi) \neq \emptyset$; existuje teda aspoň 1 ododenie $\sigma \Rightarrow_{im}^* w\xi z$, kde $w \in T^*$ a $z \in (N \cup T)^*$ je „terminujúce“
- ▶ Ak $x_1 \Rightarrow^* \varepsilon$ a $x_2 \Rightarrow^* \varepsilon$, tak $(\text{FIRST}_1(x_1 z) \cap \text{FIRST}_1(x_2 z)) \supseteq \text{FIRST}_1(z) \neq \emptyset$: spor
- ▶ Ak $x_1 \not\Rightarrow^* \varepsilon$ a $x_2 \not\Rightarrow^* \varepsilon$, tak $\text{FIRST}_1(x_1) \cap \text{FIRST}_1(x_2) \neq \emptyset$, a teda aj $\text{FIRST}_1(x_1 z) \cap \text{FIRST}_1(x_2 z) \neq \emptyset$: spor
- ▶ Ak $x_1 \not\Rightarrow^* \varepsilon$ a $x_2 \Rightarrow^* \varepsilon$, tak $\text{FIRST}_1(x_1) \cap \text{FIRST}_1(x_2) \neq \emptyset$ alebo $\text{FIRST}_1(x_1) \cap \text{FOLLOW}_1(\xi) \neq \emptyset$
 - ▶ V prvom prípade dostávame spor rovnako ako vyššie

SLL(1) vs. LL(1) na úrovni gramatík

- ▶ Každá SLL(k) gramatika je očividne aj LL(k) gramatika
- ▶ Ukážeme, že každá LL(1) gramatika je súčasne aj SLL(1) gramatika
- ▶ Sporom, nech $G = (N, T, P, \sigma)$ je LL(1), ale nie je SLL(1)

Keďže G nie je SLL(1):

- ▶ $\text{FIRST}_1(x_1 \text{FOLLOW}_1(\xi)) \cap \text{FIRST}_1(x_2 \text{FOLLOW}_1(\xi)) \neq \emptyset$
pre nejakú dvojicu rôznych pravidiel $\xi \rightarrow x_1, \xi \rightarrow x_2 \in P$
- ▶ Nutne $\text{FOLLOW}_1(\xi) \neq \emptyset$; existuje teda aspoň 1 odvozenie $\sigma \Rightarrow_{im}^* w\xi z$, kde $w \in T^*$ a $z \in (N \cup T)^*$ je „terminujúce“
- ▶ Ak $x_1 \Rightarrow^* \varepsilon$ a $x_2 \Rightarrow^* \varepsilon$, tak $(\text{FIRST}_1(x_1 z) \cap \text{FIRST}_1(x_2 z)) \supseteq \text{FIRST}_1(z) \neq \emptyset$: spor
- ▶ Ak $x_1 \not\Rightarrow^* \varepsilon$ a $x_2 \not\Rightarrow^* \varepsilon$, tak $\text{FIRST}_1(x_1) \cap \text{FIRST}_1(x_2) \neq \emptyset$, a teda aj $\text{FIRST}_1(x_1 z) \cap \text{FIRST}_1(x_2 z) \neq \emptyset$: spor
- ▶ Ak $x_1 \not\Rightarrow^* \varepsilon$ a $x_2 \Rightarrow^* \varepsilon$, tak $\text{FIRST}_1(x_1) \cap \text{FIRST}_1(x_2) \neq \emptyset$ alebo $\text{FIRST}_1(x_1) \cap \text{FOLLOW}_1(\xi) \neq \emptyset$
 - ▶ V prvom prípade dostávame spor rovnako ako vyššie
 - ▶ V druhom prípade existuje $w' \in T^*$ a $z' \in (N \cup T)^*$ tak, že $\sigma \Rightarrow_{im}^* w'\xi z'$ a $\text{FIRST}_1(x_1 z') \cap \text{FIRST}_1(x_2 z') \neq \emptyset$: spor

SLL(1) vs. LL(1) na úrovni gramatík

- ▶ Každá SLL(k) gramatika je očividne aj LL(k) gramatika
- ▶ Ukážeme, že každá LL(1) gramatika je súčasne aj SLL(1) gramatika
- ▶ Sporom, nech $G = (N, T, P, \sigma)$ je LL(1), ale nie je SLL(1)

Keďže G nie je SLL(1):

- ▶ $\text{FIRST}_1(x_1 \text{ FOLLOW}_1(\xi)) \cap \text{FIRST}_1(x_2 \text{ FOLLOW}_1(\xi)) \neq \emptyset$
pre nejakú dvojicu rôznych pravidiel $\xi \rightarrow x_1, \xi \rightarrow x_2 \in P$
- ▶ Nutne $\text{FOLLOW}_1(\xi) \neq \emptyset$; existuje teda aspoň 1 odvozenie $\sigma \Rightarrow_{im}^* w\xi z$, kde $w \in T^*$ a $z \in (N \cup T)^*$ je „terminujúce“
- ▶ Ak $x_1 \Rightarrow^* \varepsilon$ a $x_2 \Rightarrow^* \varepsilon$, tak $(\text{FIRST}_1(x_1 z) \cap \text{FIRST}_1(x_2 z)) \supseteq \text{FIRST}_1(z) \neq \emptyset$: spor
- ▶ Ak $x_1 \not\Rightarrow^* \varepsilon$ a $x_2 \not\Rightarrow^* \varepsilon$, tak $\text{FIRST}_1(x_1) \cap \text{FIRST}_1(x_2) \neq \emptyset$, a teda aj $\text{FIRST}_1(x_1 z) \cap \text{FIRST}_1(x_2 z) \neq \emptyset$: spor
- ▶ Ak $x_1 \not\Rightarrow^* \varepsilon$ a $x_2 \Rightarrow^* \varepsilon$, tak $\text{FIRST}_1(x_1) \cap \text{FIRST}_1(x_2) \neq \emptyset$ alebo $\text{FIRST}_1(x_1) \cap \text{FOLLOW}_1(\xi) \neq \emptyset$
 - ▶ V prvom prípade dostávame spor rovnako ako vyššie
 - ▶ V druhom prípade existuje $w' \in T^*$ a $z' \in (N \cup T)^*$ tak, že $\sigma \Rightarrow_{im}^* w'\xi z'$ a $\text{FIRST}_1(x_1 z') \cap \text{FIRST}_1(x_2 z') \neq \emptyset$: spor
- ▶ Ak $x_1 \Rightarrow^* \varepsilon$ a $x_2 \not\Rightarrow^* \varepsilon$, argumentácia je symetrická

SLL(2) vs. LL(2) na úrovni gramatík

Uvažujme bezkontextovou gramatiku G s následujícími pravidly:

SLL(2) vs. LL(2) na úrovni gramatík

Uvažujme bezkontextovú gramatiku G s nasledujúcimi pravidlami:

$$\sigma \rightarrow a\alpha ab \mid b\alpha bb$$

$$\alpha \rightarrow a \mid \varepsilon$$

SLL(2) vs. LL(2) na úrovni gramatík

Uvažujme bezkontextovú gramatiku G s nasledujúcimi pravidlami:

$$\sigma \rightarrow a\alpha ab \mid b\alpha bb$$

$$\alpha \rightarrow a \mid \varepsilon$$

- Neterminál σ očividne „nebude robiť problémy“

SLL(2) vs. LL(2) na úrovni gramatík

Uvažujme bezkontextovú gramatiku G s nasledujúcimi pravidlami:

$$\sigma \rightarrow a\alpha ab \mid b\alpha bb$$

$$\alpha \rightarrow a \mid \varepsilon$$

- Neterminál σ očividne „nebude robiť problémy“

Zjavne existujú iba dve vetné formy v G obsahujúce α :

SLL(2) vs. LL(2) na úrovni gramatík

Uvažujme bezkontextovú gramatiku G s nasledujúcimi pravidlami:

$$\sigma \rightarrow a\alpha ab \mid b\alpha bb$$

$$\alpha \rightarrow a \mid \varepsilon$$

- ▶ Neterminál σ očividne „nebude robiť problémy“

Zjavne existujú iba dve vetné formy v G obsahujúce α :

- ▶ $\sigma \Rightarrow_{lm} a\alpha ab$

SLL(2) vs. LL(2) na úrovni gramatík

Uvažujme bezkontextovú gramatiku G s nasledujúcimi pravidlami:

$$\sigma \rightarrow a\alpha ab \mid b\alpha bb$$

$$\alpha \rightarrow a \mid \varepsilon$$

- Neterminál σ očividne „nebude robiť problémy“

Zjavne existujú iba dve vetné formy v G obsahujúce α :

- $\sigma \Rightarrow_{lm} a\alpha ab$
 - $\text{FIRST}_2(aab) = \{aa\}$, $\text{FIRST}_2(\varepsilon ab) = \{ab\}$

SLL(2) vs. LL(2) na úrovni gramatík

Uvažujme bezkontextovú gramatiku G s nasledujúcimi pravidlami:

$$\sigma \rightarrow a\alpha ab \mid b\alpha bb$$

$$\alpha \rightarrow a \mid \varepsilon$$

- ▶ Neterminál σ očividne „nebude robiť problémy“

Zjavne existujú iba dve vetné formy v G obsahujúce α :

- ▶ $\sigma \Rightarrow_{lm} a\alpha ab$
 - ▶ $\text{FIRST}_2(aab) = \{aa\}$, $\text{FIRST}_2(\varepsilon ab) = \{ab\}$
 - ▶ $\{aa\} \cap \{ab\} = \emptyset$

SLL(2) vs. LL(2) na úrovni gramatík

Uvažujme bezkontextovú gramatiku G s nasledujúcimi pravidlami:

$$\sigma \rightarrow a\alpha ab \mid b\alpha bb$$

$$\alpha \rightarrow a \mid \varepsilon$$

- ▶ Neterminál σ očividne „nebude robiť problémy“

Zjavne existujú iba dve vetné formy v G obsahujúce α :

- ▶ $\sigma \Rightarrow_{lm} a\alpha ab$
 - ▶ $\text{FIRST}_2(aab) = \{aa\}$, $\text{FIRST}_2(\varepsilon ab) = \{ab\}$
 - ▶ $\{aa\} \cap \{ab\} = \emptyset$
- ▶ $\sigma \Rightarrow_{lm} b\alpha bb$

SLL(2) vs. LL(2) na úrovni gramatík

Uvažujme bezkontextovú gramatiku G s nasledujúcimi pravidlami:

$$\sigma \rightarrow a\alpha ab \mid b\alpha bb$$

$$\alpha \rightarrow a \mid \varepsilon$$

- ▶ Neterminál σ očividne „nebude robiť problémy“

Zjavne existujú iba dve vetné formy v G obsahujúce α :

- ▶ $\sigma \Rightarrow_{lm} a\alpha ab$
 - ▶ $\text{FIRST}_2(aab) = \{aa\}$, $\text{FIRST}_2(\varepsilon ab) = \{ab\}$
 - ▶ $\{aa\} \cap \{ab\} = \emptyset$
- ▶ $\sigma \Rightarrow_{lm} b\alpha bb$
 - ▶ $\text{FIRST}(abb) = \{ab\}$, $\text{FIRST}(\varepsilon bb) = \{bb\}$

SLL(2) vs. LL(2) na úrovni gramatík

Uvažujme bezkontextovú gramatiku G s nasledujúcimi pravidlami:

$$\sigma \rightarrow a\alpha ab \mid b\alpha bb$$

$$\alpha \rightarrow a \mid \varepsilon$$

- ▶ Neterminál σ očividne „nebude robiť problémy“

Zjavne existujú iba dve vetné formy v G obsahujúce α :

- ▶ $\sigma \Rightarrow_{lm} a\alpha ab$
 - ▶ $\text{FIRST}_2(aab) = \{aa\}$, $\text{FIRST}_2(\varepsilon ab) = \{ab\}$
 - ▶ $\{aa\} \cap \{ab\} = \emptyset$
- ▶ $\sigma \Rightarrow_{lm} b\alpha bb$
 - ▶ $\text{FIRST}(abb) = \{ab\}$, $\text{FIRST}(\varepsilon bb) = \{bb\}$
 - ▶ $\{ab\} \cap \{bb\} = \emptyset$

SLL(2) vs. LL(2) na úrovni gramatík

Uvažujme bezkontextovú gramatiku G s nasledujúcimi pravidlami:

$$\sigma \rightarrow a\alpha ab \mid b\alpha bb$$

$$\alpha \rightarrow a \mid \varepsilon$$

- ▶ Neterminál σ očividne „nebude robiť problémy“

Zjavne existujú iba dve vetné formy v G obsahujúce α :

- ▶ $\sigma \Rightarrow_{lm} a\alpha ab$
 - ▶ $\text{FIRST}_2(aab) = \{aa\}$, $\text{FIRST}_2(\varepsilon ab) = \{ab\}$
 - ▶ $\{aa\} \cap \{ab\} = \emptyset$
- ▶ $\sigma \Rightarrow_{lm} b\alpha bb$
 - ▶ $\text{FIRST}(abb) = \{ab\}$, $\text{FIRST}(\varepsilon bb) = \{bb\}$
 - ▶ $\{ab\} \cap \{bb\} = \emptyset$
- ▶ Gramatika je LL(2)

SLL(2) vs. LL(2) na úrovni gramatík

Uvažujme bezkontextovú gramatiku G s nasledujúcimi pravidlami:

$$\sigma \rightarrow a\alpha ab \mid b\alpha bb$$

$$\alpha \rightarrow a \mid \varepsilon$$

- ▶ Neterminál σ očividne „nebude robiť problémy“

Zjavne existujú iba dve vetné formy v G obsahujúce α :

- ▶ $\sigma \Rightarrow_{lm} a\alpha ab$
 - ▶ $\text{FIRST}_2(aab) = \{aa\}$, $\text{FIRST}_2(\varepsilon ab) = \{ab\}$
 - ▶ $\{aa\} \cap \{ab\} = \emptyset$
- ▶ $\sigma \Rightarrow_{lm} b\alpha bb$
 - ▶ $\text{FIRST}(abb) = \{ab\}$, $\text{FIRST}(\varepsilon bb) = \{bb\}$
 - ▶ $\{ab\} \cap \{bb\} = \emptyset$
- ▶ Gramatika je LL(2)

Zjavne ale tiež:

SLL(2) vs. LL(2) na úrovni gramatík

Uvažujme bezkontextovú gramatiku G s nasledujúcimi pravidlami:

$$\sigma \rightarrow a\alpha ab \mid b\alpha bb$$

$$\alpha \rightarrow a \mid \varepsilon$$

- ▶ Neterminál σ očividne „nebude robiť problémy“

Zjavne existujú iba dve vetné formy v G obsahujúce α :

- ▶ $\sigma \Rightarrow_{lm} a\alpha ab$
 - ▶ $\text{FIRST}_2(aab) = \{aa\}$, $\text{FIRST}_2(\varepsilon ab) = \{ab\}$
 - ▶ $\{aa\} \cap \{ab\} = \emptyset$
- ▶ $\sigma \Rightarrow_{lm} b\alpha bb$
 - ▶ $\text{FIRST}(abb) = \{ab\}$, $\text{FIRST}(\varepsilon bb) = \{bb\}$
 - ▶ $\{ab\} \cap \{bb\} = \emptyset$
- ▶ Gramatika je LL(2)

Zjavne ale tiež:

- ▶ $\text{FOLLOW}_2(\alpha) = \{ab, bb\}$

SLL(2) vs. LL(2) na úrovni gramatík

Uvažujme bezkontextovú gramatiku G s nasledujúcimi pravidlami:

$$\sigma \rightarrow a\alpha ab \mid b\alpha bb$$

$$\alpha \rightarrow a \mid \varepsilon$$

- ▶ Neterminál σ očividne „nebude robiť problémy“

Zjavne existujú iba dve vetné formy v G obsahujúce α :

- ▶ $\sigma \Rightarrow_{lm} a\alpha ab$
 - ▶ $\text{FIRST}_2(aab) = \{aa\}$, $\text{FIRST}_2(\varepsilon ab) = \{ab\}$
 - ▶ $\{aa\} \cap \{ab\} = \emptyset$
- ▶ $\sigma \Rightarrow_{lm} b\alpha bb$
 - ▶ $\text{FIRST}(abb) = \{ab\}$, $\text{FIRST}(\varepsilon bb) = \{bb\}$
 - ▶ $\{ab\} \cap \{bb\} = \emptyset$
- ▶ Gramatika je LL(2)

Zjavne ale tiež:

- ▶ $\text{FOLLOW}_2(\alpha) = \{ab, bb\}$
- ▶ $\text{FIRST}_2(a\text{FOLLOW}_2(\alpha)) = \{aa, ab\}$

SLL(2) vs. LL(2) na úrovni gramatík

Uvažujme bezkontextovú gramatiku G s nasledujúcimi pravidlami:

$$\sigma \rightarrow a\alpha ab \mid b\alpha bb$$

$$\alpha \rightarrow a \mid \varepsilon$$

- ▶ Neterminál σ očividne „nebude robiť problémy“

Zjavne existujú iba dve vetné formy v G obsahujúce α :

- ▶ $\sigma \Rightarrow_{lm} a\alpha ab$
 - ▶ $\text{FIRST}_2(aab) = \{aa\}$, $\text{FIRST}_2(\varepsilon ab) = \{ab\}$
 - ▶ $\{aa\} \cap \{ab\} = \emptyset$
- ▶ $\sigma \Rightarrow_{lm} b\alpha bb$
 - ▶ $\text{FIRST}(abb) = \{ab\}$, $\text{FIRST}(\varepsilon bb) = \{bb\}$
 - ▶ $\{ab\} \cap \{bb\} = \emptyset$
- ▶ Gramatika je LL(2)

Zjavne ale tiež:

- ▶ $\text{FOLLOW}_2(\alpha) = \{ab, bb\}$
- ▶ $\text{FIRST}_2(a\text{FOLLOW}_2(\alpha)) = \{aa, ab\}$
- ▶ $\text{FIRST}_2(\varepsilon\text{FOLLOW}_2(\alpha)) = \{ab, bb\}$

SLL(2) vs. LL(2) na úrovni gramatík

Uvažujme bezkontextovú gramatiku G s nasledujúcimi pravidlami:

$$\sigma \rightarrow a\alpha ab \mid b\alpha bb$$

$$\alpha \rightarrow a \mid \varepsilon$$

- ▶ Neterminál σ očividne „nebude robiť problémy“

Zjavne existujú iba dve vetné formy v G obsahujúce α :

- ▶ $\sigma \Rightarrow_{lm} a\alpha ab$
 - ▶ $\text{FIRST}_2(a\alpha b) = \{aa\}$, $\text{FIRST}_2(\varepsilon\alpha b) = \{ab\}$
 - ▶ $\{aa\} \cap \{ab\} = \emptyset$
- ▶ $\sigma \Rightarrow_{lm} b\alpha bb$
 - ▶ $\text{FIRST}(\alpha bb) = \{ab\}$, $\text{FIRST}(\varepsilon bb) = \{bb\}$
 - ▶ $\{ab\} \cap \{bb\} = \emptyset$
- ▶ Gramatika je LL(2)

Zjavne ale tiež:

- ▶ $\text{FOLLOW}_2(\alpha) = \{ab, bb\}$
- ▶ $\text{FIRST}_2(a\text{FOLLOW}_2(\alpha)) = \{aa, ab\}$
- ▶ $\text{FIRST}_2(\varepsilon\text{FOLLOW}_2(\alpha)) = \{ab, bb\}$
- ▶ $\{aa, ab\} \cap \{ab, bb\} \neq \emptyset$

SLL(2) vs. LL(2) na úrovni gramatík

Uvažujme bezkontextovú gramatiku G s nasledujúcimi pravidlami:

$$\sigma \rightarrow a\alpha ab \mid b\alpha bb$$

$$\alpha \rightarrow a \mid \varepsilon$$

- ▶ Neterminál σ očividne „nebude robiť problémy“

Zjavne existujú iba dve vetné formy v G obsahujúce α :

- ▶ $\sigma \Rightarrow_{lm} a\alpha ab$
 - ▶ $\text{FIRST}_2(aab) = \{aa\}$, $\text{FIRST}_2(\varepsilon ab) = \{ab\}$
 - ▶ $\{aa\} \cap \{ab\} = \emptyset$
- ▶ $\sigma \Rightarrow_{lm} b\alpha bb$
 - ▶ $\text{FIRST}(abb) = \{ab\}$, $\text{FIRST}(\varepsilon bb) = \{bb\}$
 - ▶ $\{ab\} \cap \{bb\} = \emptyset$
- ▶ Gramatika je LL(2)

Zjavne ale tiež:

- ▶ $\text{FOLLOW}_2(\alpha) = \{ab, bb\}$
- ▶ $\text{FIRST}_2(a\text{FOLLOW}_2(\alpha)) = \{aa, ab\}$
- ▶ $\text{FIRST}_2(\varepsilon\text{FOLLOW}_2(\alpha)) = \{ab, bb\}$
- ▶ $\{aa, ab\} \cap \{ab, bb\} \neq \emptyset$
- ▶ Gramatika nie je SLL(2)

Parsovanie SLL(k) gramatík

Bez dôkazu budeme využívať fakt, že:

Parsovanie SLL(k) gramatík

Bez dôkazu budeme využívať fakt, že:

- ▶ Množiny $\text{FIRST}_k(x)$ a $\text{FOLLOW}_k(x)$ sú efektívne skonštruovateľné

Parsovanie SLL(k) gramatík

Bez dôkazu budeme využívať fakt, že:

- Množiny $\text{FIRST}_k(x)$ a $\text{FOLLOW}_k(x)$ sú efektívne skonštruovateľné

Pre každé $\xi \in N$ a $u \in T^{\leq k}$, nech $T(\xi, u)$ je:

Parsovanie SLL(k) gramatík

Bez dôkazu budeme využívať fakt, že:

- ▶ Množiny $\text{FIRST}_k(x)$ a $\text{FOLLOW}_k(x)$ sú efektívne skonštruovateľné

Pre každé $\xi \in N$ a $u \in T^{\leq k}$, nech $T(\xi, u)$ je:

- ▶ Jediné pravidlo $\xi \rightarrow x$ také, že $u \in \text{FIRST}_k(x \text{ FOLLOW}_k(\xi))$
(ak také pravidlo existuje)

Parsovanie SLL(k) gramatík

Bez dôkazu budeme využívať fakt, že:

- ▶ Množiny $\text{FIRST}_k(x)$ a $\text{FOLLOW}_k(x)$ sú efektívne skonštruovateľné

Pre každé $\xi \in N$ a $u \in T^{\leq k}$, nech $T(\xi, u)$ je:

- ▶ Jediné pravidlo $\xi \rightarrow x$ také, že $u \in \text{FIRST}_k(x \text{ FOLLOW}_k(\xi))$
(ak také pravidlo existuje)
- ▶ Nedefinovaná hodnota, ak také pravidlo neexistuje

Parsovanie SLL(k) gramatík

Bez dôkazu budeme využívať fakt, že:

- ▶ Množiny $\text{FIRST}_k(x)$ a $\text{FOLLOW}_k(x)$ sú efektívne skonštruovateľné

Pre každé $\xi \in N$ a $u \in T^{\leq k}$, nech $T(\xi, u)$ je:

- ▶ Jediné pravidlo $\xi \rightarrow x$ také, že $u \in \text{FIRST}_k(x \text{ FOLLOW}_k(\xi))$
(ak také pravidlo existuje)
- ▶ Nedefinovaná hodnota, ak také pravidlo neexistuje
- ▶ $T(\cdot, \cdot)$ – parsovacia tabuľka

Parsovanie SLL(k) gramatík

Bez dôkazu budeme využívať fakt, že:

- ▶ Množiny $\text{FIRST}_k(x)$ a $\text{FOLLOW}_k(x)$ sú efektívne skonštruovateľné

Pre každé $\xi \in N$ a $u \in T^{\leq k}$, nech $T(\xi, u)$ je:

- ▶ Jediné pravidlo $\xi \rightarrow x$ také, že $u \in \text{FIRST}_k(x \text{ FOLLOW}_k(\xi))$
(ak také pravidlo existuje)
- ▶ Nedefinovaná hodnota, ak také pravidlo neexistuje
- ▶ $T(\cdot, \cdot)$ – parsovacia tabuľka

Parsovací algoritmus:

Parovanie SLL(k) gramatík

Bez dôkazu budeme využívať fakt, že:

- ▶ Množiny $\text{FIRST}_k(x)$ a $\text{FOLLOW}_k(x)$ sú efektívne skonštruovateľné

Pre každé $\xi \in N$ a $u \in T^{\leq k}$, nech $T(\xi, u)$ je:

- ▶ Jediné pravidlo $\xi \rightarrow x$ také, že $u \in \text{FIRST}_k(x \text{ FOLLOW}_k(\xi))$
(ak také pravidlo existuje)
- ▶ Nedefinovaná hodnota, ak také pravidlo neexistuje
- ▶ $T(\cdot, \cdot)$ – parovacia tabuľka

Parovací algoritmus:

- ▶ Ak je na vrchu zásobníka terminál c , vymaž ho zo zásobníka
a prečítaj c zo vstupu

Parovanie SLL(k) gramatík

Bez dôkazu budeme využívať fakt, že:

- ▶ Množiny $\text{FIRST}_k(x)$ a $\text{FOLLOW}_k(x)$ sú efektívne skonštruovateľné

Pre každé $\xi \in N$ a $u \in T^{\leq k}$, nech $T(\xi, u)$ je:

- ▶ Jediné pravidlo $\xi \rightarrow x$ také, že $u \in \text{FIRST}_k(x \text{ FOLLOW}_k(\xi))$
(ak také pravidlo existuje)
- ▶ Nedefinovaná hodnota, ak také pravidlo neexistuje
- ▶ $T(\cdot, \cdot)$ – parovacia tabuľka

Parovací algoritmus:

- ▶ Ak je na vrchu zásobníka terminál c , vymaž ho zo zásobníka a prečítaj c zo vstupu
- ▶ Ak je zásobník prázdny, akceptuj

Parovanie SLL(k) gramatík

Bez dôkazu budeme využívať fakt, že:

- ▶ Množiny $\text{FIRST}_k(x)$ a $\text{FOLLOW}_k(x)$ sú efektívne skonštruovateľné

Pre každé $\xi \in N$ a $u \in T^{\leq k}$, nech $T(\xi, u)$ je:

- ▶ Jediné pravidlo $\xi \rightarrow x$ také, že $u \in \text{FIRST}_k(x \text{ FOLLOW}_k(\xi))$ (ak také pravidlo existuje)
- ▶ Nedefinovaná hodnota, ak také pravidlo neexistuje
- ▶ $T(\cdot, \cdot)$ – parovacia tabuľka

Parovací algoritmus:

- ▶ Ak je na vrchu zásobníka terminál c , vymaž ho zo zásobníka a prečítaj c zo vstupu
- ▶ Ak je zásobník prázdny, akceptuj
- ▶ Inak musí byť na vrchu zásobníka neterminál ξ

Parovanie SLL(k) gramatík

Bez dôkazu budeme využívať fakt, že:

- ▶ Množiny $\text{FIRST}_k(x)$ a $\text{FOLLOW}_k(x)$ sú efektívne skonštruovateľné

Pre každé $\xi \in N$ a $u \in T^{\leq k}$, nech $T(\xi, u)$ je:

- ▶ Jediné pravidlo $\xi \rightarrow x$ také, že $u \in \text{FIRST}_k(x \text{ FOLLOW}_k(\xi))$ (ak také pravidlo existuje)
- ▶ Nedefinovaná hodnota, ak také pravidlo neexistuje
- ▶ $T(\cdot, \cdot)$ – parovacia tabuľka

Parovací algoritmus:

- ▶ Ak je na vrchu zásobníka terminál c , vymaž ho zo zásobníka a prečítaj c zo vstupu
- ▶ Ak je zásobník prázdny, akceptuj
- ▶ Inak musí byť na vrchu zásobníka neterminál ξ
- ▶ Nech $u = k:w$, kde w je nedočítaný vstup

Parsovanie SLL(k) gramatík

Bez dôkazu budeme využívať fakt, že:

- ▶ Množiny $\text{FIRST}_k(x)$ a $\text{FOLLOW}_k(x)$ sú efektívne skonštruovateľné

Pre každé $\xi \in N$ a $u \in T^{\leq k}$, nech $T(\xi, u)$ je:

- ▶ Jediné pravidlo $\xi \rightarrow x$ také, že $u \in \text{FIRST}_k(x \text{ FOLLOW}_k(\xi))$ (ak také pravidlo existuje)
- ▶ Nedefinovaná hodnota, ak také pravidlo neexistuje
- ▶ $T(\cdot, \cdot)$ – parsovacia tabuľka

Parsovací algoritmus:

- ▶ Ak je na vrchu zásobníka terminál c , vymaž ho zo zásobníka a prečítaj c zo vstupu
- ▶ Ak je zásobník prázdny, akceptuj
- ▶ Inak musí byť na vrchu zásobníka neterminál ξ
- ▶ Nech $u = k:w$, kde w je nedočítaný vstup
- ▶ Vymaž ξ a nahraď ho reverzom pravej strany pravidla $T(\xi, u)$

Parovanie SLL(k) gramatík

Bez dôkazu budeme využívať fakt, že:

- ▶ Množiny $\text{FIRST}_k(x)$ a $\text{FOLLOW}_k(x)$ sú efektívne skonštruovateľné

Pre každé $\xi \in N$ a $u \in T^{\leq k}$, nech $T(\xi, u)$ je:

- ▶ Jediné pravidlo $\xi \rightarrow x$ také, že $u \in \text{FIRST}_k(x \text{ FOLLOW}_k(\xi))$ (ak také pravidlo existuje)
- ▶ Nedefinovaná hodnota, ak také pravidlo neexistuje
- ▶ $T(\cdot, \cdot)$ – parovacia tabuľka

Parovací algoritmus:

- ▶ Ak je na vrchu zásobníka terminál c , vymaž ho zo zásobníka a prečítaj c zo vstupu
- ▶ Ak je zásobník prázdny, akceptuj
- ▶ Inak musí byť na vrchu zásobníka neterminál ξ
- ▶ Nech $u = k:w$, kde w je nedočítaný vstup
- ▶ Vymaž ξ a nahraď ho reverzom pravej strany pravidla $T(\xi, u)$

Parovanie LL(k) gramatík je o niečo komplikovanejšie

Porovnania na úrovni generovaných jazykov

Uved'me len dve tvrdenia bez dôkazu:

Porovnania na úrovni generovaných jazykov

Uved'me len dve tvrdenia bez dôkazu:

Veta

Jazyk L je generovaný $LL(k)$ gramatikou práve vtedy, keď je generovaný $SLL(k)$ gramatikou.

Porovnania na úrovni generovaných jazykov

Uved'me len dve tvrdenia bez dôkazu:

Veta

Jazyk L je generovaný $LL(k)$ gramatikou práve vtedy, keď je generovaný $SLL(k)$ gramatikou.

Veta

Pre každé k existuje jazyk L_k , ktorý je generovaný $LL(k+1)$ gramatikou, ale nie je generovaný $LL(k)$ gramatikou.

Odkazy na literatúru

V nasledujúcich knihách možno nájsť vynechané detaily:

Odkazy na literatúru

V nasledujúcich knihách možno nájsť vynechané detaily:

- ▶ Aho, A. V., Ullman, J. D.:
The Theory of Parsing, Translation, and Compiling, vol. I
Englewood Cliffs : Prentice Hall, 1972.

Odkazy na literatúru

V nasledujúcich knihách možno nájsť vynechané detaily:

- ▶ Aho, A. V., Ullman, J. D.:
The Theory of Parsing, Translation, and Compiling, vol. I
Englewood Cliffs : Prentice Hall, 1972.
- ▶ Aho, A. V., Ullman, J. D.:
The Theory of Parsing, Translation, and Compiling, vol. II
Englewood Cliffs : Prentice Hall, 1973.

Odkazy na literatúru

V nasledujúcich knihách možno nájsť vynechané detaily:

- ▶ Aho, A. V., Ullman, J. D.:
The Theory of Parsing, Translation, and Compiling, vol. I
Englewood Cliffs : Prentice Hall, 1972.
- ▶ Aho, A. V., Ullman, J. D.:
The Theory of Parsing, Translation, and Compiling, vol. II
Englewood Cliffs : Prentice Hall, 1973.
- ▶ Sippu, S., Soisalon-Soininen, E.:
Parsing Theory, vol. I
Berlin, Heidelberg : Springer, 1988.

Odkazy na literatúru

V nasledujúcich knihách možno nájsť vynechané detaily:

- ▶ Aho, A. V., Ullman, J. D.:
The Theory of Parsing, Translation, and Compiling, vol. I
Englewood Cliffs : Prentice Hall, 1972.
- ▶ Aho, A. V., Ullman, J. D.:
The Theory of Parsing, Translation, and Compiling, vol. II
Englewood Cliffs : Prentice Hall, 1973.
- ▶ Sippu, S., Soisalon-Soininen, E.:
Parsing Theory, vol. I
Berlin, Heidelberg : Springer, 1988.
- ▶ Sippu, S., Soisalon-Soininen, E.:
Parsing Theory, vol. II
Berlin, Heidelberg : Springer, 1990.

Časť 3:

Nutné podmienky na splnenie definície $LL(k)$ gramatík

Motivácia

- ▶ Existujú (aj jednoznačné) bezkontextové jazyky, ktoré nie sú generované $LL(k)$ gramatikou pre žiadne k

Motivácia

- ▶ Existujú (aj jednoznačné) bezkontextové jazyky, ktoré nie sú generované $LL(k)$ gramatikou pre žiadne k
- ▶ V praxi často stačí málo na transformáciu „zlej“ gramatiky na „dobrú“

Motivácia

- ▶ Existujú (aj jednoznačné) bezkontextové jazyky, ktoré nie sú generované $LL(k)$ gramatikou pre žiadne k
- ▶ V praxi často stačí málo na transformáciu „zlej“ gramatiky na „dobrú“
- ▶ Pri návrhu programovacích jazykov sa objavujú určité „typické problematické konštrukcie“

Motivácia

- ▶ Existujú (aj jednoznačné) bezkontextové jazyky, ktoré nie sú generované $LL(k)$ gramatikou pre žiadne k
- ▶ V praxi často stačí málo na transformáciu „zlej“ gramatiky na „dobrú“
- ▶ Pri návrhu programovacích jazykov sa objavujú určité „typické problematické konštrukcie“
- ▶ Neprítomnosť týchto „problematických konštrukcií“ v gramatike je nutnou podmienkou splnenia definície $LL(k)$ gramatiky (a často aj aplikovateľnosti iných metód syntaktickej analýzy zhora nadol)

Ľavá rekurzia a $LL(k)$ gramatiky

Neterminál ξ v bezkontextovej gramatike $G = (N, T, P, \sigma)$ je:

Ľavá rekurzia a $LL(k)$ gramatiky

Neterminál ξ v bezkontextovej gramatike $G = (N, T, P, \sigma)$ je:

- Ľavo rekurzívny, ak existuje $x \in (N \cup T)^*$ také, že $\xi \Rightarrow^+ \xi x$

Ľavá rekurzia a $LL(k)$ gramatiky

Neterminál ξ v bezkontextovej gramatike $G = (N, T, P, \sigma)$ je:

- ▶ Ľavo rekurzívny, ak existuje $x \in (N \cup T)^*$ také, že $\xi \Rightarrow^+ \xi x$
- ▶ Bezprostredne ľavo rekurzívny, ak existuje $x \in (N \cup T)^*$ také, že $\xi \rightarrow \xi x \in P$

Ľavá rekurzia a $LL(k)$ gramatiky

Neterminál ξ v bezkontextovej gramatike $G = (N, T, P, \sigma)$ je:

- ▶ Ľavo rekurzívny, ak existuje $x \in (N \cup T)^*$ také, že $\xi \Rightarrow^+ \xi x$
- ▶ Bezprostredne ľavo rekurzívny, ak existuje $x \in (N \cup T)^*$ také, že $\xi \rightarrow \xi x \in P$

Žiadna redukovaná $LL(k)$ gramatika nemôže obsahovať ľavo rekurzívny neterminál:

Ľavá rekurzia a $LL(k)$ gramatiky

Neterminál ξ v bezkontextovej gramatike $G = (N, T, P, \sigma)$ je:

- ▶ Ľavo rekurzívny, ak existuje $x \in (N \cup T)^*$ také, že $\xi \Rightarrow^+ \xi x$
- ▶ Bezprostredne ľavo rekurzívny, ak existuje $x \in (N \cup T)^*$ také, že $\xi \rightarrow \xi x \in P$

Žiadna redukovaná $LL(k)$ gramatika nemôže obsahovať ľavo rekurzívny neterminál:

- ▶ Nech $\xi \Rightarrow^+ \xi x$

Ľavá rekurzia a $LL(k)$ gramatiky

Neterminál ξ v bezkontextovej gramatike $G = (N, T, P, \sigma)$ je:

- ▶ Ľavo rekurzívny, ak existuje $x \in (N \cup T)^*$ také, že $\xi \Rightarrow^+ \xi x$
- ▶ Bezprostredne ľavo rekurzívny, ak existuje $x \in (N \cup T)^*$ také, že $\xi \rightarrow \xi x \in P$

Žiadna redukovaná $LL(k)$ gramatika nemôže obsahovať ľavo rekurzívny neterminál:

- ▶ Nech $\xi \Rightarrow^+ \xi x$
- ▶ Musí existovať aj nejaké pravidlo $\xi \rightarrow y$, kde y **nezačína** neterminálom ξ

Ľavá rekurzia a $LL(k)$ gramatiky

Neterminál ξ v bezkontextovej gramatike $G = (N, T, P, \sigma)$ je:

- ▶ Ľavo rekurzívny, ak existuje $x \in (N \cup T)^*$ také, že $\xi \Rightarrow^+ \xi x$
- ▶ Bezprostredne ľavo rekurzívny, ak existuje $x \in (N \cup T)^*$ také, že $\xi \rightarrow \xi x \in P$

Žiadna redukovaná $LL(k)$ gramatika nemôže obsahovať ľavo rekurzívny neterminál:

- ▶ Nech $\xi \Rightarrow^+ \xi x$
- ▶ Musí existovať aj nejaké pravidlo $\xi \rightarrow y$, kde y **nezačína** neterminálom ξ
- ▶ Buď $x \Rightarrow^* \varepsilon$, alebo pre všetky $u \in T^*$ také, že $x^k \Rightarrow^* u$ platí $|u| \geq k$

Ľavá rekurzia a $LL(k)$ gramatiky

Neterminál ξ v bezkontextovej gramatike $G = (N, T, P, \sigma)$ je:

- ▶ Ľavo rekurzívny, ak existuje $x \in (N \cup T)^*$ také, že $\xi \Rightarrow^+ \xi x$
- ▶ Bezprostredne ľavo rekurzívny, ak existuje $x \in (N \cup T)^*$ také, že $\xi \rightarrow \xi x \in P$

Žiadna redukovaná $LL(k)$ gramatika nemôže obsahovať ľavo rekurzívny neterminál:

- ▶ Nech $\xi \Rightarrow^+ \xi x$
- ▶ Musí existovať aj nejaké pravidlo $\xi \rightarrow y$, kde y **nezačína** neterminálom ξ
- ▶ Buď $x \Rightarrow^* \varepsilon$, alebo pre všetky $u \in T^*$ také, že $x^k \Rightarrow^* u$ platí $|u| \geq k$
- ▶ Existujú $w \in T^*$ a $z \in (N \cup T)^*$ také, že $\sigma \Rightarrow_{lm}^* w \xi x^k z$

Ľavá rekurzia a $LL(k)$ gramatiky

Neterminál ξ v bezkontextovej gramatike $G = (N, T, P, \sigma)$ je:

- ▶ Ľavo rekurzívny, ak existuje $x \in (N \cup T)^*$ také, že $\xi \Rightarrow^+ \xi x$
- ▶ Bezprostredne ľavo rekurzívny, ak existuje $x \in (N \cup T)^*$ také, že $\xi \rightarrow \xi x \in P$

Žiadna redukovaná $LL(k)$ gramatika nemôže obsahovať ľavo rekurzívny neterminál:

- ▶ Nech $\xi \Rightarrow^+ \xi x$
- ▶ Musí existovať aj nejaké pravidlo $\xi \rightarrow y$, kde y **nezačína** neterminálom ξ
- ▶ Buď $x \Rightarrow^* \varepsilon$, alebo pre všetky $u \in T^*$ také, že $x^k \Rightarrow^* u$ platí $|u| \geq k$
- ▶ Existujú $w \in T^*$ a $z \in (N \cup T)^*$ také, že $\sigma \Rightarrow_{lm}^* w \xi x^k z$
- ▶ $\emptyset \neq \text{FIRST}_k(yx^kz) \subseteq \text{FIRST}_k(\xi x x^k z)$

Ľavá rekurzia a $LL(k)$ gramatiky

Neterminál ξ v bezkontextovej gramatike $G = (N, T, P, \sigma)$ je:

- ▶ Ľavo rekurzívny, ak existuje $x \in (N \cup T)^*$ také, že $\xi \Rightarrow^+ \xi x$
- ▶ Bezprostredne ľavo rekurzívny, ak existuje $x \in (N \cup T)^*$ také, že $\xi \rightarrow \xi x \in P$

Žiadna redukovaná $LL(k)$ gramatika nemôže obsahovať ľavo rekurzívny neterminál:

- ▶ Nech $\xi \Rightarrow^+ \xi x$
- ▶ Musí existovať aj nejaké pravidlo $\xi \rightarrow y$, kde y **nezačína** neterminálom ξ
- ▶ Buď $x \Rightarrow^* \varepsilon$, alebo pre všetky $u \in T^*$ také, že $x^k \Rightarrow^* u$ platí $|u| \geq k$
- ▶ Existujú $w \in T^*$ a $z \in (N \cup T)^*$ také, že $\sigma \Rightarrow_{lm}^* w \xi x^k z$
- ▶ $\emptyset \neq \text{FIRST}_k(yx^kz) \subseteq \text{FIRST}_k(\xi x x^k z)$

Ľavá rekurzia znemožňuje napríklad aj rekurzívny zostup

Odstraňovanie ľavej rekurzie

- ▶ Nech G je bezkontextová gramatika

Odstraňovanie ľavej rekurzie

- ▶ Nech G je bezkontextová gramatika
- ▶ Je možné algoritmicky zostrojiť bezkontextovú gramatiku G' bez ľavej rekurzie takú, že $L(G') = L(G)$

Odstraňovanie ľavej rekurzie

- ▶ Nech G je bezkontextová gramatika
- ▶ Je možné algoritmicky zostrojiť bezkontextovú gramatiku G' bez ľavej rekurzie takú, že $L(G') = L(G)$
- ▶ Podobné ako prvá fáza prevodu gramatiky do Greibachovej normálneho tvaru (Formálne jazyky a automaty 2)

Odstraňovanie bezprostrednej ľavej rekurzie

- Nech ξ je ľavo rekurzívny neterminál v $G = (N, T, P, \sigma)$

Odstraňovanie bezprostrednej ľavej rekurzie

- ▶ Nech ξ je ľavo rekurzívny neterminál v $G = (N, T, P, \sigma)$
- ▶ Nech $\xi \rightarrow \xi x_1 \mid \xi x_2 \mid \dots \mid \xi x_r \in P$

Odstraňovanie bezprostrednej ľavej rekurzie

- ▶ Nech ξ je ľavo rekurzívny neterminál v $G = (N, T, P, \sigma)$
- ▶ Nech $\xi \rightarrow \xi x_1 \mid \xi x_2 \mid \dots \mid \xi x_r \in P$
- ▶ BUNV, nech $x_1, \dots, x_r$ sú neprázdne

Odstraňovanie bezprostrednej ľavej rekurzie

- ▶ Nech ξ je ľavo rekurzívny neterminál v $G = (N, T, P, \sigma)$
- ▶ Nech $\xi \rightarrow \xi x_1 \mid \xi x_2 \mid \dots \mid \xi x_r \in P$
- ▶ BUNV, nech $x_1, \dots, x_r$ sú neprázdne
- ▶ Nech $\xi \rightarrow w_1 \mid w_2 \mid \dots \mid w_s$ sú všetky zvyšné pravidlá

Odstraňovanie bezprostrednej ľavej rekurzie

- ▶ Nech ξ je ľavo rekurzívny neterminál v $G = (N, T, P, \sigma)$
- ▶ Nech $\xi \rightarrow \xi x_1 \mid \xi x_2 \mid \dots \mid \xi x_r \in P$
- ▶ BUNV, nech $x_1, \dots, x_r$ sú neprázdne
- ▶ Nech $\xi \rightarrow w_1 \mid w_2 \mid \dots \mid w_s$ sú všetky zvyšné pravidlá

Tieto pravidlá nahradíme novými pravidlami (kde ξ' je nový neterminál):

Odstraňovanie bezprostrednej ľavej rekurzie

- ▶ Nech ξ je ľavo rekurzívny neterminál v $G = (N, T, P, \sigma)$
- ▶ Nech $\xi \rightarrow \xi x_1 \mid \xi x_2 \mid \dots \mid \xi x_r \in P$
- ▶ BUNV, nech $x_1, \dots, x_r$ sú neprázdne
- ▶ Nech $\xi \rightarrow w_1 \mid w_2 \mid \dots \mid w_s$ sú všetky zvyšné pravidlá

Tieto pravidlá nahradíme novými pravidlami (kde ξ' je nový neterminál):

- ▶ $\xi \rightarrow w_1 \xi' \mid w_2 \xi' \mid \dots \mid w_s \xi'$

Odstraňovanie bezprostrednej ľavej rekurzie

- ▶ Nech ξ je ľavo rekurzívny neterminál v $G = (N, T, P, \sigma)$
- ▶ Nech $\xi \rightarrow \xi x_1 \mid \xi x_2 \mid \dots \mid \xi x_r \in P$
- ▶ BUNV, nech $x_1, \dots, x_r$ sú neprázdne
- ▶ Nech $\xi \rightarrow w_1 \mid w_2 \mid \dots \mid w_s$ sú všetky zvyšné pravidlá

Tieto pravidlá nahradíme novými pravidlami (kde ξ' je nový neterminál):

- ▶ $\xi \rightarrow w_1 \xi' \mid w_2 \xi' \mid \dots \mid w_s \xi'$
- ▶ $\xi' \rightarrow x_1 \xi' \mid x_2 \xi' \mid \dots \mid x_r \xi' \mid \varepsilon$

Príklad: gramatika pre aritmetické výrazy

Prvý pokus:

Príklad: gramatika pre aritmetické výrazy

Prvý pokus:

$$\begin{aligned}\langle \text{výraz} \rangle \rightarrow & \langle \text{výraz} \rangle \cdot \langle \text{výraz} \rangle \mid \langle \text{výraz} \rangle / \langle \text{výraz} \rangle \mid \\ & \langle \text{výraz} \rangle + \langle \text{výraz} \rangle \mid \langle \text{výraz} \rangle - \langle \text{výraz} \rangle \mid \\ & (\langle \text{výraz} \rangle) \mid \mathbf{num}\end{aligned}$$

Príklad: gramatika pre aritmetické výrazy

Prvý pokus:

$$\begin{aligned}\langle \text{výraz} \rangle \rightarrow & \langle \text{výraz} \rangle \cdot \langle \text{výraz} \rangle \mid \langle \text{výraz} \rangle / \langle \text{výraz} \rangle \mid \\ & \langle \text{výraz} \rangle + \langle \text{výraz} \rangle \mid \langle \text{výraz} \rangle - \langle \text{výraz} \rangle \mid \\ & (\langle \text{výraz} \rangle) \mid \mathbf{num}\end{aligned}$$

- ▶ Táto gramatika je korektná, ale viacznačná

Príklad: gramatika pre aritmetické výrazy

Prvý pokus:

$$\begin{aligned}\langle \text{výraz} \rangle \rightarrow & \langle \text{výraz} \rangle \cdot \langle \text{výraz} \rangle \mid \langle \text{výraz} \rangle / \langle \text{výraz} \rangle \mid \\ & \langle \text{výraz} \rangle + \langle \text{výraz} \rangle \mid \langle \text{výraz} \rangle - \langle \text{výraz} \rangle \mid \\ & (\langle \text{výraz} \rangle) \mid \mathbf{num}\end{aligned}$$

- ▶ Táto gramatika je korektná, ale viacznačná
- ▶ Na nájdenie správneho stromu odvodu treba pridať priority

Príklad: gramatika pre aritmetické výrazy

Prvý pokus:

$$\begin{aligned}\langle \text{výraz} \rangle \rightarrow & \langle \text{výraz} \rangle \cdot \langle \text{výraz} \rangle \mid \langle \text{výraz} \rangle / \langle \text{výraz} \rangle \mid \\ & \langle \text{výraz} \rangle + \langle \text{výraz} \rangle \mid \langle \text{výraz} \rangle - \langle \text{výraz} \rangle \mid \\ & (\langle \text{výraz} \rangle) \mid \mathbf{num}\end{aligned}$$

- ▶ Táto gramatika je korektná, ale viacznačná
- ▶ Na nájdenie správneho stromu odvodenia treba pridať priority
- ▶ Napríklad ANTLR4 si vie poradiť aj s gramatikou uvedenou vyššie (je ale dôležité chápať tomu, čo sa tam deje)

Príklad: gramatika pre aritmetické výrazy

Prvý pokus:

$$\begin{aligned}\langle \text{výraz} \rangle \rightarrow & \langle \text{výraz} \rangle \cdot \langle \text{výraz} \rangle \mid \langle \text{výraz} \rangle / \langle \text{výraz} \rangle \mid \\ & \langle \text{výraz} \rangle + \langle \text{výraz} \rangle \mid \langle \text{výraz} \rangle - \langle \text{výraz} \rangle \mid \\ & (\langle \text{výraz} \rangle) \mid \mathbf{num}\end{aligned}$$

- ▶ Táto gramatika je korektná, ale viacznačná
- ▶ Na nájdenie správneho stromu odvodu treba pridať priority
- ▶ Napríklad ANTLR4 si vie poradiť aj s gramatikou uvedenou vyššie (je ale dôležité chápať tomu, čo sa tam deje)
- ▶ Skúsme radšej nájsť jednoznačnú gramatiku...

Príklad: gramatika pre aritmetické výrazy

Druhý pokus:

Príklad: gramatika pre aritmetické výrazy

Druhý pokus:

$$\langle \text{výraz} \rangle \rightarrow \langle \text{výraz} \rangle + \langle \text{člen} \rangle \mid \langle \text{výraz} \rangle - \langle \text{člen} \rangle \mid \langle \text{člen} \rangle$$
$$\langle \text{člen} \rangle \rightarrow \langle \text{člen} \rangle \cdot \langle \text{faktor} \rangle \mid \langle \text{člen} \rangle / \langle \text{faktor} \rangle \mid \langle \text{faktor} \rangle$$
$$\langle \text{faktor} \rangle \rightarrow (\langle \text{výraz} \rangle) \mid \mathbf{num}$$

Príklad: gramatika pre aritmetické výrazy

Druhý pokus:

$$\langle \text{výraz} \rangle \rightarrow \langle \text{výraz} \rangle + \langle \text{člen} \rangle \mid \langle \text{výraz} \rangle - \langle \text{člen} \rangle \mid \langle \text{člen} \rangle$$
$$\langle \text{člen} \rangle \rightarrow \langle \text{člen} \rangle \cdot \langle \text{faktor} \rangle \mid \langle \text{člen} \rangle / \langle \text{faktor} \rangle \mid \langle \text{faktor} \rangle$$
$$\langle \text{faktor} \rangle \rightarrow (\langle \text{výraz} \rangle) \mid \mathbf{num}$$

- ▶ Táto gramatika je korektná a jednoznačná, ale ľavo rekurzívna

Príklad: gramatika pre aritmetické výrazy

Druhý pokus:

$$\langle \text{výraz} \rangle \rightarrow \langle \text{výraz} \rangle + \langle \text{člen} \rangle \mid \langle \text{výraz} \rangle - \langle \text{člen} \rangle \mid \langle \text{člen} \rangle$$
$$\langle \text{člen} \rangle \rightarrow \langle \text{člen} \rangle \cdot \langle \text{faktor} \rangle \mid \langle \text{člen} \rangle / \langle \text{faktor} \rangle \mid \langle \text{faktor} \rangle$$
$$\langle \text{faktor} \rangle \rightarrow (\langle \text{výraz} \rangle) \mid \mathbf{num}$$

- ▶ Táto gramatika je korektná a jednoznačná, ale ľavo rekurzívna
- ▶ Obsahuje iba bezprostrednú ľavú rekurziu

Príklad: gramatika pre aritmetické výrazy

Druhý pokus:

$$\langle \text{výraz} \rangle \rightarrow \langle \text{výraz} \rangle + \langle \text{člen} \rangle \mid \langle \text{výraz} \rangle - \langle \text{člen} \rangle \mid \langle \text{člen} \rangle$$
$$\langle \text{člen} \rangle \rightarrow \langle \text{člen} \rangle \cdot \langle \text{faktor} \rangle \mid \langle \text{člen} \rangle / \langle \text{faktor} \rangle \mid \langle \text{faktor} \rangle$$
$$\langle \text{faktor} \rangle \rightarrow (\langle \text{výraz} \rangle) \mid \mathbf{num}$$

- ▶ Táto gramatika je korektná a jednoznačná, ale ľavo rekurzívna
- ▶ Obsahuje iba bezprostrednú ľavú rekurziu
- ▶ Aplikujme preto konštrukciu na odstránenie bezprostrednej ľavej rekurzie

Príklad: gramatika pre aritmetické výrazy

Pôvodná gramatika:

Príklad: gramatika pre aritmetické výrazy

Pôvodná gramatika:

$$\langle \text{výraz} \rangle \rightarrow \langle \text{výraz} \rangle + \langle \text{člen} \rangle \mid \langle \text{výraz} \rangle - \langle \text{člen} \rangle \mid \langle \text{člen} \rangle$$
$$\langle \text{člen} \rangle \rightarrow \langle \text{člen} \rangle \cdot \langle \text{faktor} \rangle \mid \langle \text{člen} \rangle / \langle \text{faktor} \rangle \mid \langle \text{faktor} \rangle$$
$$\langle \text{faktor} \rangle \rightarrow (\langle \text{výraz} \rangle) \mid \mathbf{num}$$

Príklad: gramatika pre aritmetické výrazy

Pôvodná gramatika:

$$\langle \text{výraz} \rangle \rightarrow \langle \text{výraz} \rangle + \langle \text{člen} \rangle \mid \langle \text{výraz} \rangle - \langle \text{člen} \rangle \mid \langle \text{člen} \rangle$$
$$\langle \text{člen} \rangle \rightarrow \langle \text{člen} \rangle \cdot \langle \text{faktor} \rangle \mid \langle \text{člen} \rangle / \langle \text{faktor} \rangle \mid \langle \text{faktor} \rangle$$
$$\langle \text{faktor} \rangle \rightarrow (\langle \text{výraz} \rangle) \mid \mathbf{num}$$

Výsledná gramatika:

Príklad: gramatika pre aritmetické výrazy

Pôvodná gramatika:

$$\langle \text{výraz} \rangle \rightarrow \langle \text{výraz} \rangle + \langle \text{člen} \rangle \mid \langle \text{výraz} \rangle - \langle \text{člen} \rangle \mid \langle \text{člen} \rangle$$
$$\langle \text{člen} \rangle \rightarrow \langle \text{člen} \rangle \cdot \langle \text{faktor} \rangle \mid \langle \text{člen} \rangle / \langle \text{faktor} \rangle \mid \langle \text{faktor} \rangle$$
$$\langle \text{faktor} \rangle \rightarrow (\langle \text{výraz} \rangle) \mid \mathbf{num}$$

Výsledná gramatika:

$$\langle \text{výraz} \rangle \rightarrow \langle \text{člen} \rangle \langle \text{výraz} \rangle'$$
$$\langle \text{výraz} \rangle' \rightarrow + \langle \text{člen} \rangle \langle \text{výraz} \rangle' \mid - \langle \text{člen} \rangle \langle \text{výraz} \rangle' \mid \varepsilon$$
$$\langle \text{člen} \rangle \rightarrow \langle \text{faktor} \rangle \langle \text{člen} \rangle'$$
$$\langle \text{člen} \rangle' \rightarrow \cdot \langle \text{faktor} \rangle \langle \text{člen} \rangle' \mid / \langle \text{faktor} \rangle \langle \text{člen} \rangle' \mid \varepsilon$$
$$\langle \text{faktor} \rangle \rightarrow (\langle \text{výraz} \rangle) \mid \mathbf{num}$$

Príklad: gramatika pre aritmetické výrazy

Pôvodná gramatika:

$$\begin{aligned}\langle \text{výraz} \rangle &\rightarrow \langle \text{výraz} \rangle + \langle \text{člen} \rangle \mid \langle \text{výraz} \rangle - \langle \text{člen} \rangle \mid \langle \text{člen} \rangle \\ \langle \text{člen} \rangle &\rightarrow \langle \text{člen} \rangle \cdot \langle \text{faktor} \rangle \mid \langle \text{člen} \rangle / \langle \text{faktor} \rangle \mid \langle \text{faktor} \rangle \\ \langle \text{faktor} \rangle &\rightarrow (\langle \text{výraz} \rangle) \mid \mathbf{num}\end{aligned}$$

Výsledná gramatika:

$$\begin{aligned}\langle \text{výraz} \rangle &\rightarrow \langle \text{člen} \rangle \langle \text{výraz} \rangle' \\ \langle \text{výraz} \rangle' &\rightarrow + \langle \text{člen} \rangle \langle \text{výraz} \rangle' \mid - \langle \text{člen} \rangle \langle \text{výraz} \rangle' \mid \varepsilon \\ \langle \text{člen} \rangle &\rightarrow \langle \text{faktor} \rangle \langle \text{člen} \rangle' \\ \langle \text{člen} \rangle' &\rightarrow \cdot \langle \text{faktor} \rangle \langle \text{člen} \rangle' \mid / \langle \text{faktor} \rangle \langle \text{člen} \rangle' \mid \varepsilon \\ \langle \text{faktor} \rangle &\rightarrow (\langle \text{výraz} \rangle) \mid \mathbf{num}\end{aligned}$$

- Pôvodná gramatika zodpovedá ľavo asociatívnym operátorom:
napríklad $1 - 2 - 3 = (1 - 2) - 3$

Príklad: gramatika pre aritmetické výrazy

Pôvodná gramatika:

$$\langle \text{výraz} \rangle \rightarrow \langle \text{výraz} \rangle + \langle \text{člen} \rangle \mid \langle \text{výraz} \rangle - \langle \text{člen} \rangle \mid \langle \text{člen} \rangle$$
$$\langle \text{člen} \rangle \rightarrow \langle \text{člen} \rangle \cdot \langle \text{faktor} \rangle \mid \langle \text{člen} \rangle / \langle \text{faktor} \rangle \mid \langle \text{faktor} \rangle$$
$$\langle \text{faktor} \rangle \rightarrow (\langle \text{výraz} \rangle) \mid \mathbf{num}$$

Výsledná gramatika:

$$\langle \text{výraz} \rangle \rightarrow \langle \text{člen} \rangle \langle \text{výraz} \rangle'$$
$$\langle \text{výraz} \rangle' \rightarrow + \langle \text{člen} \rangle \langle \text{výraz} \rangle' \mid - \langle \text{člen} \rangle \langle \text{výraz} \rangle' \mid \varepsilon$$
$$\langle \text{člen} \rangle \rightarrow \langle \text{faktor} \rangle \langle \text{člen} \rangle'$$
$$\langle \text{člen} \rangle' \rightarrow \cdot \langle \text{faktor} \rangle \langle \text{člen} \rangle' \mid / \langle \text{faktor} \rangle \langle \text{člen} \rangle' \mid \varepsilon$$
$$\langle \text{faktor} \rangle \rightarrow (\langle \text{výraz} \rangle) \mid \mathbf{num}$$

- ▶ Pôvodná gramatika zodpovedá ľavo asociatívnym operátorom:
napríklad $1 - 2 - 3 = (1 - 2) - 3$
- ▶ V novej gramatike už syntaktický strom nezodpovedá očakávanému aritmetickému stromu

Príklad: gramatika pre aritmetické výrazy

Pôvodná gramatika:

$$\langle \text{výraz} \rangle \rightarrow \langle \text{výraz} \rangle + \langle \text{člen} \rangle \mid \langle \text{výraz} \rangle - \langle \text{člen} \rangle \mid \langle \text{člen} \rangle$$
$$\langle \text{člen} \rangle \rightarrow \langle \text{člen} \rangle \cdot \langle \text{faktor} \rangle \mid \langle \text{člen} \rangle / \langle \text{faktor} \rangle \mid \langle \text{faktor} \rangle$$
$$\langle \text{faktor} \rangle \rightarrow (\langle \text{výraz} \rangle) \mid \mathbf{num}$$

Výsledná gramatika:

$$\langle \text{výraz} \rangle \rightarrow \langle \text{člen} \rangle \langle \text{výraz} \rangle'$$
$$\langle \text{výraz} \rangle' \rightarrow + \langle \text{člen} \rangle \langle \text{výraz} \rangle' \mid - \langle \text{člen} \rangle \langle \text{výraz} \rangle' \mid \varepsilon$$
$$\langle \text{člen} \rangle \rightarrow \langle \text{faktor} \rangle \langle \text{člen} \rangle'$$
$$\langle \text{člen} \rangle' \rightarrow \cdot \langle \text{faktor} \rangle \langle \text{člen} \rangle' \mid / \langle \text{faktor} \rangle \langle \text{člen} \rangle' \mid \varepsilon$$
$$\langle \text{faktor} \rangle \rightarrow (\langle \text{výraz} \rangle) \mid \mathbf{num}$$

- ▶ Pôvodná gramatika zodpovedá ľavo asociatívnym operátorom:
napríklad $1 - 2 - 3 = (1 - 2) - 3$
- ▶ V novej gramatike už syntaktický strom nezodpovedá očakávanému aritmetickému stromu
- ▶ Časté riešenie: napísanie procedúr na vyrobenie „abstraktného syntaktického stromu“

Odstraňovanie ľavej rekurzie

Predpokladajme, že gramatika na vstupe:

Odstraňovanie ľavej rekurzie

Predpokladajme, že gramatika na vstupe:

- ▶ Je „bezepsilonová“

Odstraňovanie ľavej rekurzie

Predpokladajme, že gramatika na vstupe:

- ▶ Je „bezepsilonová“
- ▶ Neobsahuje „reťazové pravidlá“ ($\alpha \rightarrow \beta$)

Odstraňovanie ľavej rekurzie

Predpokladajme, že gramatika na vstupe:

- ▶ Je „bezepsilonová“
- ▶ Neobsahuje „reťazové pravidlá“ ($\alpha \rightarrow \beta$)
- ▶ Má neterminály $\xi_1, \dots, \xi_n$ (BUNV)

Odstraňovanie ľavej rekurzie

Predpokladajme, že gramatika na vstupe:

- ▶ Je „bezepsilonová“
- ▶ Neobsahuje „reťazové pravidlá“ ($\alpha \rightarrow \beta$)
- ▶ Má neterminály $\xi_1, \dots, \xi_n$ (BUNV)

Ľavú rekurziu možno odstrániť nasledujúcim algoritmom:

Odstraňovanie ľavej rekurzie

Predpokladajme, že gramatika na vstupe:

- ▶ Je „bezepsilonová“
- ▶ Neobsahuje „reťazové pravidlá“ ($\alpha \rightarrow \beta$)
- ▶ Má neterminály $\xi_1, \dots, \xi_n$ (BUNV)

Ľavú rekurziu možno odstrániť nasledujúcim algoritmom:

1. Opakuj pre $k = 1, \dots, n$:

Odstraňovanie ľavej rekurzie

Predpokladajme, že gramatika na vstupe:

- ▶ Je „bezepsilonová“
- ▶ Neobsahuje „reťazové pravidlá“ ($\alpha \rightarrow \beta$)
- ▶ Má neterminály $\xi_1, \dots, \xi_n$ (BUNV)

Ľavú rekurziu možno odstrániť nasledujúcim algoritmom:

1. Opakuj pre $k = 1, \dots, n$:
 - 1.1. Opakuj pre $j = 1, \dots, k - 1$:

Odstraňovanie ľavej rekurzie

Predpokladajme, že gramatika na vstupe:

- ▶ Je „bezepsilonová“
- ▶ Neobsahuje „reťazové pravidlá“ ($\alpha \rightarrow \beta$)
- ▶ Má neterminály $\xi_1, \dots, \xi_n$ (BUNV)

Ľavú rekurziu možno odstrániť nasledujúcim algoritmom:

1. Opakuj pre $k = 1, \dots, n$:
 - 1.1. Opakuj pre $j = 1, \dots, k - 1$:
 - 1.1.1. Nech $\xi_j \rightarrow w_1 \mid \dots \mid w_m$ sú všetky pravidlá zo ξ_j

Odstraňovanie ľavej rekurzie

Predpokladajme, že gramatika na vstupe:

- ▶ Je „bezepsilonová“
- ▶ Neobsahuje „reťazové pravidlá“ ($\alpha \rightarrow \beta$)
- ▶ Má neterminály $\xi_1, \dots, \xi_n$ (BUNV)

Ľavú rekurziu možno odstrániť nasledujúcim algoritmom:

1. Opakuj pre $k = 1, \dots, n$:

1.1. Opakuj pre $j = 1, \dots, k - 1$:

1.1.1. Nech $\xi_j \rightarrow w_1 \mid \dots \mid w_m$ sú všetky pravidlá zo ξ_j

1.1.2. Nahraď každé pravidlo $\xi_k \rightarrow \xi_j x$ pravidlami $\xi_k \rightarrow w_1 x \mid \dots \mid w_m x$

Odstraňovanie ľavej rekurzie

Predpokladajme, že gramatika na vstupe:

- ▶ Je „bezepsilonová“
- ▶ Neobsahuje „reťazové pravidlá“ ($\alpha \rightarrow \beta$)
- ▶ Má neterminály $\xi_1, \dots, \xi_n$ (BUNV)

Ľavú rekurziu možno odstrániť nasledujúcim algoritmom:

1. Opakuj pre $k = 1, \dots, n$:
 - 1.1. Opakuj pre $j = 1, \dots, k - 1$:
 - 1.1.1. Nech $\xi_j \rightarrow w_1 \mid \dots \mid w_m$ sú všetky pravidlá zo ξ_j
 - 1.1.2. Nahraď každé pravidlo $\xi_k \rightarrow \xi_j x$ pravidlami $\xi_k \rightarrow w_1 x \mid \dots \mid w_m x$
 - 1.2. Odstráň bezprostrednú ľavú rekurziu v ξ_k

Odstraňovanie ľavej rekurzie

Predpokladajme, že gramatika na vstupe:

- ▶ Je „bezepsilonová“
- ▶ Neobsahuje „reťazové pravidlá“ ($\alpha \rightarrow \beta$)
- ▶ Má neterminály $\xi_1, \dots, \xi_n$ (BUNV)

Ľavú rekurziu možno odstrániť nasledujúcim algoritmom:

1. Opakuj pre $k = 1, \dots, n$:
 - 1.1. Opakuj pre $j = 1, \dots, k - 1$:
 - 1.1.1. Nech $\xi_j \rightarrow w_1 \mid \dots \mid w_m$ sú všetky pravidlá zo ξ_j
 - 1.1.2. Nahraď každé pravidlo $\xi_k \rightarrow \xi_j x$ pravidlami $\xi_k \rightarrow w_1 x \mid \dots \mid w_m x$
 - 1.2. Odstráň bezprostrednú ľavú rekurziu v ξ_k
- ▶ Právě strany pravidiel zo ξ_k teraz začínajú buď terminálom, alebo neterminálom ξ_j , kde $j > k$

Odstraňovanie ľavej rekurzie

Predpokladajme, že gramatika na vstupe:

- ▶ Je „bezepsilonová“
- ▶ Neobsahuje „reťazové pravidlá“ ($\alpha \rightarrow \beta$)
- ▶ Má neterminály $\xi_1, \dots, \xi_n$ (BUNV)

Ľavú rekurziu možno odstrániť nasledujúcim algoritmom:

1. Opakuj pre $k = 1, \dots, n$:

1.1. Opakuj pre $j = 1, \dots, k - 1$:

1.1.1. Nech $\xi_j \rightarrow w_1 \mid \dots \mid w_m$ sú všetky pravidlá zo ξ_j

1.1.2. Nahraď každé pravidlo $\xi_k \rightarrow \xi_j x$ pravidlami $\xi_k \rightarrow w_1 x \mid \dots \mid w_m x$

1.2. Odstráň bezprostrednú ľavú rekurziu v ξ_k

- ▶ Pravé strany pravidiel zo ξ_k teraz začínajú buď terminálom, alebo neterminálom ξ_j , kde $j > k$
- ▶ Pravé strany pravidiel zo ξ_k' sú takisto „v poriadku“

Odstraňovanie ľavej rekurzie (príklad)

Uvažujme gramatiku s nasledujúcimi pravidlami:

Odstraňovanie ľavej rekurzie (príklad)

Uvažujme gramatiku s nasledujúcimi pravidlami:

$$\xi_1 \rightarrow \xi_1 a \mid \xi_2 \xi_2$$

$$\xi_2 \rightarrow \xi_1 \xi_1 \mid \xi_2 \xi_2 \mid b$$

Odstraňovanie ľavej rekurzie (príklad)

Uvažujme gramatiku s nasledujúcimi pravidlami:

$$\xi_1 \rightarrow \xi_1 a \mid \xi_2 \xi_2$$

$$\xi_2 \rightarrow \xi_1 \xi_1 \mid \xi_2 \xi_2 \mid b$$

$k = 1$, bod 1.2:

Odstraňovanie ľavej rekurzie (príklad)

Uvažujme gramatiku s nasledujúcimi pravidlami:

$$\xi_1 \rightarrow \xi_1 a \mid \xi_2 \xi_2$$

$$\xi_2 \rightarrow \xi_1 \xi_1 \mid \xi_2 \xi_2 \mid b$$

$k = 1$, bod 1.2:

$$\xi_1 \rightarrow \xi_2 \xi_2 \xi'_1$$

$$\xi'_1 \rightarrow a \xi'_1 \mid \varepsilon$$

$$\xi_2 \rightarrow \xi_1 \xi_1 \mid \xi_2 \xi_2 \mid b$$

Odstraňovanie ľavej rekurzie (príklad)

Uvažujme gramatiku s nasledujúcimi pravidlami:

$$\xi_1 \rightarrow \xi_1 a \mid \xi_2 \xi_2$$

$$\xi_2 \rightarrow \xi_1 \xi_1 \mid \xi_2 \xi_2 \mid b$$

$k = 1$, bod 1.2:

$$\xi_1 \rightarrow \xi_2 \xi_2 \xi'_1$$

$$\xi'_1 \rightarrow a \xi'_1 \mid \varepsilon$$

$$\xi_2 \rightarrow \xi_1 \xi_1 \mid \xi_2 \xi_2 \mid b$$

$k = 2$, bod 1.1:

Odstraňovanie ľavej rekurzie (príklad)

Uvažujme gramatiku s nasledujúcimi pravidlami:

$$\xi_1 \rightarrow \xi_1 a \mid \xi_2 \xi_2$$

$$\xi_2 \rightarrow \xi_1 \xi_1 \mid \xi_2 \xi_2 \mid b$$

$k = 1$, bod 1.2:

$$\xi_1 \rightarrow \xi_2 \xi_2 \xi'_1$$

$$\xi'_1 \rightarrow a \xi'_1 \mid \varepsilon$$

$$\xi_2 \rightarrow \xi_1 \xi_1 \mid \xi_2 \xi_2 \mid b$$

$k = 2$, bod 1.1:

$$\xi_1 \rightarrow \xi_2 \xi_2 \xi'_1$$

$$\xi'_1 \rightarrow a \xi'_1 \mid \varepsilon$$

$$\xi_2 \rightarrow \xi_2 \xi_2 \xi'_1 \xi_1 \mid \xi_2 \xi_2 \mid b$$

Odstraňovanie ľavej rekurzie (príklad)

Uvažujme gramatiku s nasledujúcimi pravidlami:

$$\xi_1 \rightarrow \xi_1 a \mid \xi_2 \xi_2$$

$$\xi_2 \rightarrow \xi_1 \xi_1 \mid \xi_2 \xi_2 \mid b$$

$k = 1$, bod 1.2:

$$\xi_1 \rightarrow \xi_2 \xi_2 \xi'_1$$

$$\xi'_1 \rightarrow a \xi'_1 \mid \varepsilon$$

$$\xi_2 \rightarrow \xi_1 \xi_1 \mid \xi_2 \xi_2 \mid b$$

$k = 2$, bod 1.1:

$$\xi_1 \rightarrow \xi_2 \xi_2 \xi'_1$$

$$\xi'_1 \rightarrow a \xi'_1 \mid \varepsilon$$

$$\xi_2 \rightarrow \xi_2 \xi_2 \xi'_1 \xi_1 \mid \xi_2 \xi_2 \mid b$$

$k = 2$, bod 1.2:

Odstraňovanie ľavej rekurzie (príklad)

Uvažujme gramatiku s nasledujúcimi pravidlami:

$$\xi_1 \rightarrow \xi_1 a \mid \xi_2 \xi_2$$

$$\xi_2 \rightarrow \xi_1 \xi_1 \mid \xi_2 \xi_2 \mid b$$

$k = 1$, bod 1.2:

$$\xi_1 \rightarrow \xi_2 \xi_2 \xi'_1$$

$$\xi'_1 \rightarrow a \xi'_1 \mid \varepsilon$$

$$\xi_2 \rightarrow \xi_1 \xi_1 \mid \xi_2 \xi_2 \mid b$$

$k = 2$, bod 1.1:

$$\xi_1 \rightarrow \xi_2 \xi_2 \xi'_1$$

$$\xi'_1 \rightarrow a \xi'_1 \mid \varepsilon$$

$$\xi_2 \rightarrow \xi_2 \xi_2 \xi'_1 \xi_1 \mid \xi_2 \xi_2 \mid b$$

$k = 2$, bod 1.2:

$$\xi_1 \rightarrow \xi_2 \xi_2 \xi'_1$$

$$\xi'_1 \rightarrow a \xi'_1 \mid \varepsilon$$

$$\xi_2 \rightarrow b \xi'_2$$

$$\xi'_2 \rightarrow \xi_2 \xi'_1 \xi_1 \xi'_2 \mid \xi_2 \xi'_2 \mid \varepsilon$$

Ľavý rozklad

Uvažujme gramatiku s (okrem iného) nasledujúcimi pravidlami:

Ľavý rozklad

Uvažujme gramatiku s (okrem iného) nasledujúcimi pravidlami:

- ▶ $\xi \rightarrow xy$

Ľavý rozklad

Uvažujme gramatiku s (okrem iného) nasledujúcimi pravidlami:

- ▶ $\xi \rightarrow xy$
- ▶ $\xi \rightarrow xy'$

Ľavý rozklad

Uvažujme gramatiku s (okrem iného) nasledujúcimi pravidlami:

- ▶ $\xi \rightarrow xy$
- ▶ $\xi \rightarrow xy'$
- ▶ x, y, y' sú slová z $(N \cup T)^*$

Ľavý rozklad

Uvažujme gramatiku s (okrem iného) nasledujúcimi pravidlami:

- ▶ $\xi \rightarrow xy$
- ▶ $\xi \rightarrow xy'$
- ▶ x, y, y' sú slová z $(N \cup T)^*$
- ▶ Ak zostávajúci vstup začína slovom odvoditeľným z x , prediktívne parsovanie môže byť problém

Ľavý rozklad

Uvažujme gramatiku s (okrem iného) nasledujúcimi pravidlami:

- ▶ $\xi \rightarrow xy$
- ▶ $\xi \rightarrow xy'$
- ▶ x, y, y' sú slová z $(N \cup T)^*$
- ▶ Ak zostávajúci vstup začína slovom odvoditeľným z x , prediktívne parsovanie môže byť problém

Ľavý rozklad (angl. „left factoring“):

Ľavý rozklad

Uvažujme gramatiku s (okrem iného) nasledujúcimi pravidlami:

- ▶ $\xi \rightarrow xy$
- ▶ $\xi \rightarrow xy'$
- ▶ x, y, y' sú slová z $(N \cup T)^*$
- ▶ Ak zostávajúci vstup začína slovom odvoditeľným z x , prediktívne parsovanie môže byť problém

Ľavý rozklad (angl. „left factoring“):

- ▶ $\xi \rightarrow x\xi'$

Ľavý rozklad

Uvažujme gramatiku s (okrem iného) nasledujúcimi pravidlami:

- ▶ $\xi \rightarrow xy$
- ▶ $\xi \rightarrow xy'$
- ▶ x, y, y' sú slová z $(N \cup T)^*$
- ▶ Ak zostávajúci vstup začína slovom odvoditeľným z x , prediktívne parsovanie môže byť problém

Ľavý rozklad (angl. „left factoring“):

- ▶ $\xi \rightarrow x\xi'$
- ▶ $\xi' \rightarrow y \mid y'$

Časť 4:

Pokročilé parsovacie stratégie

LL(*) parsovanie

- ▶ Používané v nižších verziách ANTLR

LL(*) parsovanie

- ▶ Používané v nižších verziách ANTLR
- ▶ Idea: ku každému neterminálu sa vytvorí konečný automat

LL(*) parsovanie

- ▶ Používané v nižších verziách ANTLR
- ▶ Idea: ku každému neterminálu sa vytvorí konečný automat
- ▶ Na zistenie nasledujúceho pravidla sa tento automat spustí na zostávajúcom vstupe

LL(*) parsovanie

- ▶ Používané v nižších verziách ANTLR
- ▶ Idea: ku každému neterminálu sa vytvorí konečný automat
- ▶ Na zistenie nasledujúceho pravidla sa tento automat spustí na zostávajúcom vstupe
- ▶ Ak je gramatika $LL(k)$, automat spraví najviac k krokov a nájde správne pravidlo

LL(*) parsovanie

- ▶ Používané v nižších verziách ANTLR
- ▶ Idea: ku každému neterminálu sa vytvorí konečný automat
- ▶ Na zistenie nasledujúceho pravidla sa tento automat spustí na zostávajúcom vstupe
- ▶ Ak je gramatika $LL(k)$, automat spraví najviac k krokov a nájde správne pravidlo
- ▶ $LL(*)$ ale „zvládne“ aj gramatiky, ktoré nie sú $LL(k)$

LL(*) parsovanie

- ▶ Používané v nižších verziách ANTLR
- ▶ Idea: ku každému neterminálu sa vytvorí konečný automat
- ▶ Na zistenie nasledujúceho pravidla sa tento automat spustí na zostávajúcom vstupe
- ▶ Ak je gramatika $LL(k)$, automat spraví najviac k krokov a nájde správne pravidlo
- ▶ $LL(*)$ ale „zvládne“ aj gramatiky, ktoré nie sú $LL(k)$
- ▶ Vďaka sémantickým predikátom v turingovsky úplnom jazyku: (teoretická) možnosť popísať ľubovoľný rekurzívny jazyk

LL(*) parsovanie

- ▶ Používané v nižších verziách ANTLR
- ▶ Idea: ku každému neterminálu sa vytvorí konečný automat
- ▶ Na zistenie nasledujúceho pravidla sa tento automat spustí na zostávajúcom vstupe
- ▶ Ak je gramatika $LL(k)$, automat spraví najviac k krokov a nájde správne pravidlo
- ▶ $LL(*)$ ale „zvládne“ aj gramatiky, ktoré nie sú $LL(k)$
- ▶ Vďaka sémantickým predikátom v turingovsky úplnom jazyku: (teoretická) možnosť popísať ľubovoľný rekurzívny jazyk
- ▶ Jediné podstatné obmedzenie: vstupná gramatika nesmie byť ľavo rekurzívna

ALL(*) parsovanie

- ▶ Rozšírenie LL(*) používané v ANTLR4

ALL(*) parsovanie

- ▶ Rozšírenie LL(*) používané v ANTLR4
- ▶ Akceptuje aj vstupné gramatiky s bezprostrednou ľavou rekurziou

ALL(*) parsovanie

- ▶ Rozšírenie LL(*) používané v ANTLR4
- ▶ Akceptuje aj vstupné gramatiky s bezprostrednou ľavou rekurziou
- ▶ Neakceptuje gramatiky s inou ako bezprostrednou ľavou rekurziou

ALL(*) parsovanie

- ▶ Rozšírenie LL(*) používané v ANTLR4
- ▶ Akceptuje aj vstupné gramatiky s bezprostrednou ľavou rekurziou
- ▶ Neakceptuje gramatiky s inou ako bezprostrednou ľavou rekurziou
- ▶ To ale zvyčajne nie je veľký problém